Parachute Extraction of a Generic Store from a C-130; a CFD Proof of Concept

THESIS

Stephen C. Platt, Maj, USAF

AFIT/GAE/ENY/05-M17

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GAE/ENY/05-M17

Parachute Extraction of a Generic Store from a C-130; a CFD Proof of Concept

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Aeronautical Engineering

Stephen C. Platt, B.S.

Maj, USAF

March, 2005

AFIT/GAE/ENY/05-M17

Parachute Extraction of a Generic Store from a C-130; a CFD Proof of Concept

Stephen C. Platt, B.S.
Maj, USAF

Approved:

/signed/	21 Mar 2005
Milton E. Franke (Chairman)	date
/signed/	21 Mar 2005
LtCol Raymond C. Maple (Member)	date
/signed/	21 Mar 2005
Maj Richard J. McMullan (Member)	date

Abstract

This thesis encompasses a feasibility analysis of a parachute extracted generic precision guided munition (PGM) from the cargo bay of a C-130 aircraft in flight. This analysis utilizes the United States Air Force (USAF) Beggar code and incorporates full physics effects as well as aerodynamic loading assuming an inviscid aircraft and viscous store for a time-accurate solution. Both an immediate application and a time varying application of the parachute force are utilized as well as two different ordnance body styles at zero and 5 degrees angle of attack (AOA) with the store placed on centerline and offset in the cargo bay. The time accurate parachute model is based on empirical data and more closely follows the force fall off as the parachute slows down during the extraction process.

The analysis shows this concept has merit. Both store body styles were successfully extracted from the cargo bay without contacting any portion of the delivery aircraft, following a safe trajectory down and away from all of the release conditions. The extraction took an average of 1.7 seconds with the immediate application of the parachute force and approximately 2.1 seconds when the time varying model was applied. The maximum roll seen during an extraction was 13 degrees, which was the largest movement on any axis.

Acknowledgements

The author would like to thank the growers and roasters of fine Columbian coffees whose efforts directly led to the timely completion of this project.

Stephen C. Platt

Table of Contents

Page

List of Figures

List of Tables

List of Abbreviations

Parachute Extraction of a Generic Store from a C-130; a CFD Proof of Concept

I. INTRODUCTION

"Thus the highest form of generalship is to balk the enemy's plans, the next best is to prevent the junction of the enemy's forces, the next in order is to attack the enemy's army in the field, and the worst policy of all is to besiege walled cities."

-Sun Tzu, the Art of War

1.1 Historical Perspective

Insights such as these remain relevant regardless of their antiquity. In a modern context, why would we send ground forces into a country before attacking the strategic centers of gravity from afar? The evolution of warfare leads inexorably to the ability to strike from greater distance with greater precision. The last quarter of the 20th century saw the rise of PGMs that were guided by an operator from a short distance, allowing for corrections in flight due to target movement or employment errors. PGMs allowed war fighters, for the first time, to effectively strike pinpoint targets representing command and control nodes, critical infrastructure points and other "centers of gravity" type targets. While extremely useful in the visual environment, adverse weather and target properties could significantly impact the successful employment of these ordnance. The advent of precise, real-time, own-ship determination of position, velocity and acceleration signaled yet another revolution in war-fighting capability.

In US Air Force's most recent conflicts, the lion's share of strategic air strikes take place from a stealthy platform dropping a glide bomb that is guided by global positioning system (GPS) satellites. Such muntions represent the new class of ordnance known as coordinate seeking weapons (CSW). Unfortunately, both gliding PGMs and CSWs suffer from two obvious drawbacks. First, the simple proximity required to

deliver these munitions given their unpowered flight profile increases the threat to the delivery aircraft. Second, commensurate with the Air Force Strategic Planner's need to strike a large array of desired mean points of impact (DMPI), the delivery platform must also be large. But to be large and close to the target necessitates stealth meaning an incredible cost for the delivery platform. Stealth technology will never be cheap. Considering the continuing evolution of both target defensive arrays and detection technology, a constant and substantial investment is required to protect the delivery platform. Powered CSWs, however, offer an alternative and as their capabilities increase our ability to follow Sun Tzu's wise counsel increases, correspondingly.

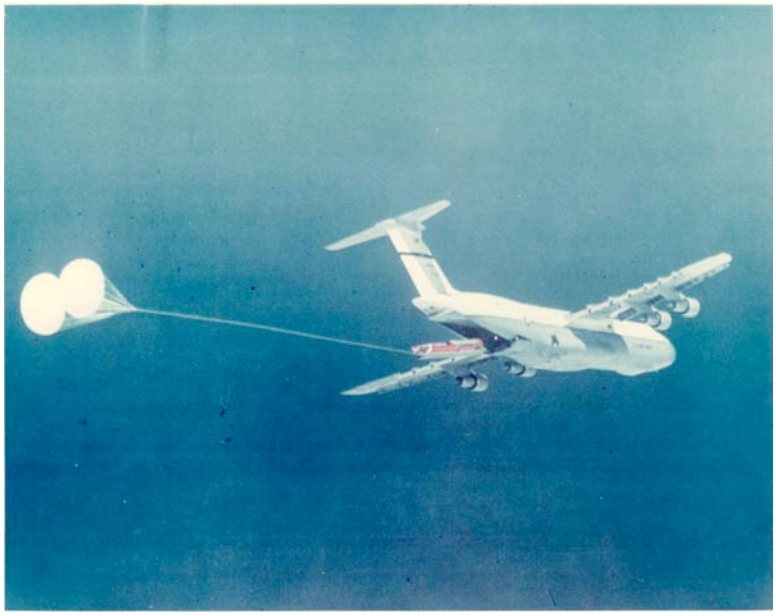
Given that these CSWs can strike a stationary target from a distance with great precision, it follows logically that the problem is now no longer proximity and its corresponding stealth requirement, but simply tonnage delivered to an appropriate launch point. Any delivery platform is suitable assuming the weapon can reach the target with the appropriate endgame energy state for weapons effects. Given this logic train, one postulated approach has been to outfit a portion of the existing cargo aircraft fleet with a large number of CSWs capable of being extracted via parachutes while in flight. By limiting the choice of munitions to those with an autonomous thrust capability, the delivery aircraft can remain relatively clear of the threat array, reducing its need for defensive hardening/threat detection, and thus cost.

If this avenue were proved feasible and exploited, the possibility to deliver a truly staggering number of munitions with pin-point accuracy to strategic targets such as the enemy's planning centers or lines of communication would become a reality. A strike of this nature could nearly end a conflict before it began, dramatically reducing the need for additional forces. Sun Tzu would be pleased.

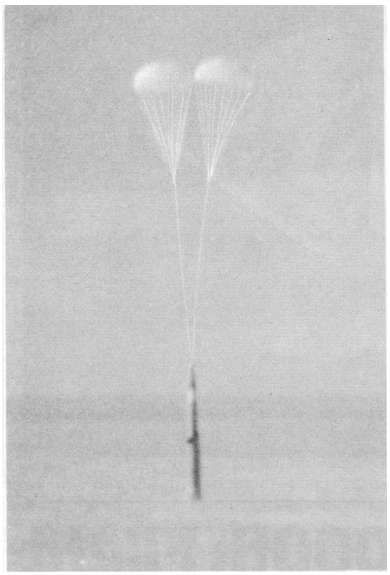
1.2 Concept Genesis

If this concept of cargo aircraft borne, parachute extracted CSWs can be proved feasible, the payoff to the war fighter would be enormous. While the extraction of a weapon from a cargo bay is not a new concept, with the Massive Ordnance

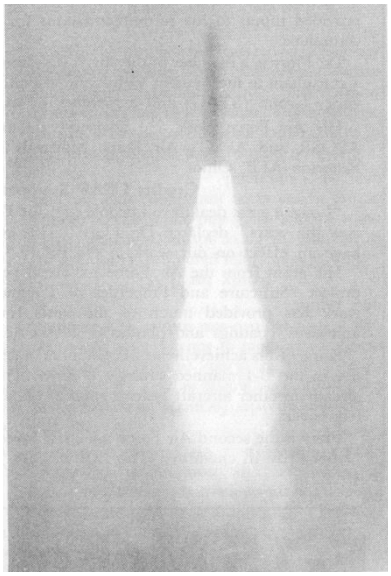
Air Blast (MOAB) deployment from a C-130(4) and an inter continental ballistic missile (ICBM) from a C-5 Galaxy(13), having a large number of these weapons with extensive real-time programming for deployment from a cargo aircraft is novel. In Figure 1 below, the C-5 deployment of an ICBM is shown(16):



(a) Deployment



(b) Under Parachutes



(c) After Successful Firing

Figure 1: Successful Deployment and Firing of an ICBM from a C-5 Galaxy

Although not addressed herein, the mounting of CSWs in the cargo bay and programming of a large number of weapons is no mean problem. In fact, a thorough Computational Fluid Dynamics (CFD) analysis of the flow field aft of a cargo aircraft with an open door and the effect of that field on deployed weapons has not been accomplished prior to this study.

One of the newest CSWs is the AGM-158 Joint Air to Surface Standoff Missile (JASSM), whose long range flight profile provides the Air Force planner with an option other than a close range strike by stealthy aircraft for a strategic target set. An extensive discussion of the vetting process leading to the JASSM selection for extraction from a cargo bay is outlined in the precursor to this analysis by Franke and Ari (5). While is was not possible to use the actual Lockheed JASSM CFD model, a generic store was created having similar ballistic and inertial properties. This approach allowed for a wider dissemination of the results while capturing the major effects of a "JASSM-like" payload during extraction from a cargo bay. After the baseline run, a "flattened" store was used, having the same inertial qualities, but reduced by 35 percent in cross-sectional height in an attempt to more closely model the aerodynamic loading on the store.

The C-130 Hercules was chosen as the delivery aircraft because of its ubiquity, service life extension program indicating its continued active status for many years and the availability of a baseline computational grid for research. The original grid was for an AC-130 that included a dense grid area on the left side to capture the blast effects of the 105mm Howitzer. Unfortunately, there was no cargo door or interior bay in this model, so the grids had to be generated and appropriate boundary conditions set for the solution. For this effort, the Air Force Seek Eagle Office (AFSEO) Computational Aeromechanics Team (CAT) was invaluable.

For this problem to be tractable at this stage, certain simplifications were necessary. Chief among these was a simplification of the parachute extraction force. The current state of the art in CFD is beginning to see full physics models of parachutes,

but their complexity is too great for this problem given the time allowed. Instead, a straight line force in the store's body frame of reference is applied to act at the center of gravity (CG) throughout the duration of the simulation. Also, the geometry of the cargo door in the model lacks the three dimensional thickness of the actual door, but has the same physical length and width.

With these conditions set, the solution proceeded with the aid of the Air Force's Beggar code for solving a full motion separation problem complete with the effects of physics and aerodynamic loading on the aircraft, cargo door and store during the separation process.

1.3 Literature Review

The paper that led directly to this effort has been discussed above and contains further rationale for the selection of a cargo aircraft as a delivery platform when paired with a powered CSW. In general, Franke and Ari discuss the rationale for the selection of the JASSM and its paring with various cargo aircraft. In addition, they propose a number of storage/carriage solutions, that have not been modeled in this paper (5).

Johnson et al. (6) examined the cargo area of a C-130 utilizing both an experimental and CFD analysis with the primary objective of examining the flow field for features that would affect the safety of airborne paratroopers upon exit from the aircraft. The experimental data were obtained with laser light sheet and dye flow visualization coupled with particle image velocimetry in a water tunnel at the U.S. Air Force Academy. Their CFD solution was developed using the COBALT code on a viscous model. In both cases, the model and its control surfaces are static in the flow field and the forces analyzed. This is an important distinction when compared to a store separation analysis; there is no provision made for body relative motion (e.g. a moving store). Of note in their results is a significant predicted updraft behind the open cargo door that could potentially affect the safe extraction of a store from the cargo bay of a C-130.

Lijewski and Belk discussed the generation of the baseline C-130 grid utilized in this analysis in their paper(10). They take the reader through a bottom up approach to CFD grid construction using the National Grid Project (NGP) grid generation code and the subsequent assembly of these grids into the entire structure using the U.S. Air Force's Beggar code. Of particular interest to the author was their background on the limitations of the overset grid method as applied to this specific aircraft. Lijewski and Belk discuss at length techniques to capture "orphan points" or those points of failed interpolation, between adjacent super blocks (groups of grids that overlap and remain static relative to one another during the solution).

Serrano et al. presented a baseline analysis of the drag of a parachute mounted behind a C-130 in their paper (15). Utilizing Detached Eddy Simulation (DES) and the COBALT code, they were able to arrive at solution in which they observed the same type of updraft directly aft of the aircraft and its effects on the parachute when placed in this region as Johnson et al. (6). Once again, the parachute was stationary relative to the C-130 and the loading was analyzed on the parachute in the turbulent flow field behind the aircraft.

Finally, an article in Air Force Research Lab (AFRL) Technology Horizons, "Massive Ordnance Air Blast Weapon Development" detailed the proof of concept of an extraction of an early version CSW from the cargo bay of a C-130 (4). The successful testing and deployment of this system lends credence to the entire effort.

II. COMPUTATIONAL APPROACH

2.1 Beggar

The U.S. Air Force's Beggar code is a finite volume, cell-centered solver that is based in a combination of Fortran and C languages using structured grids. Beggar solves the integral form of the Naviar-Stokes equations shown below as a reference:

$$\int_V \frac{\partial Q}{\partial t} dV + \int_A (F_c - F_v) dA = 0 \tag{1}$$

where

$$Q = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E_t \end{Bmatrix} \quad , \quad F_c = \begin{Bmatrix} \rho U \\ \rho U u + p n_x \\ \rho U v + p n_y \\ \rho U w + p n_z \\ (E_t + p) U \end{Bmatrix} \quad , \quad F_v = \begin{Bmatrix} 0 \\ [\tau' \cdot \hat{n}] \\ \theta_x n_x + \theta_y n_y + \theta_z n_z \end{Bmatrix} ,$$

and

$$\theta_i = u_j \tau_{ij} + k \frac{\partial T}{\partial x_i} \tag{2}$$

is the work done by the viscous stresses and the heat conduction of the fluid(3), $U = \mathbf{V} \cdot \hat{n}$ is the contravariant velocity, and τ is the stress tensor defined from the deformation law for a Newtonian fluid given by:

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \delta_{ij} \lambda \nabla \mathbf{V} \tag{3}$$

A more in depth discussion of the implementation in Beggar is available in Rizk et al.(14). Beggar was originally developed in the Air Force Wright Laboratory Armament Directorate and continues to be updated by the AFSEO CAT. Beggar incorporates a grid assembly tool, flow solver and six degree of freedom (6DOF) model (1). The strength of this marriage is in the ability to model the interaction of the flow field around the complex geometries of an aircraft in flight and stores as they move

relative to, or separate from, the aircraft, yielding a time accurate solution. Beggar was developed specifically to work in the demanding regimes of the transonic and supersonic flight profiles in which many of our combat aircraft cruise during the attack phase of flight. As opposed to the other methods discussed in the literature review, Beggar provides the user with the opportunity to see the complex flow fields around these moving bodies interact with the bodies as they move relative to one another. By incorporating all the relevant physical properties of the bodies and watching the effects of the aerodynamic loads (with the coupling of the 6DOF solver), it it possible to run a preflight simulation of many store separation problems prior to submitting aircrews to the dangers inherent in a flight test.

2.2 Overset Grids in Beggar

Beggar uses block-to-block, patched and overlapping (or "overset") grids when solving complex geometries such as aircraft and their associated stores (1). Block-to-block simply means that all the cells along the germane boundary of the current gird exactly meet with cells of the same physical dimension in an adjacent grid and the cell boundaries have the same three-space designation. Patched grids are those that still meet along the same border, but the dimensions of the adjacent cells are not equal, in that the cell edges do not line up. The difference between these two methods and the overset grid approach is best illustrated by Figure 2(1).

The overset grids approach to a solution has several inherent advantages. The first and most important during the initial phase of the problem is the ease of grid construction. The second advantage is apparent when objects are moving relative to one another during the solution time frame. Rather than having to change the grid every time step to represent the new positions of the objects, moving objects are encased in their own separate grid systems that move with each body that is in motion(11). In this problem, the cargo bay door and weapon were constructed as separate "super blocks", which is Beggar-parlance for a grouping of grids that are joined and remain stationary relative to one another. After an appropriate boundary

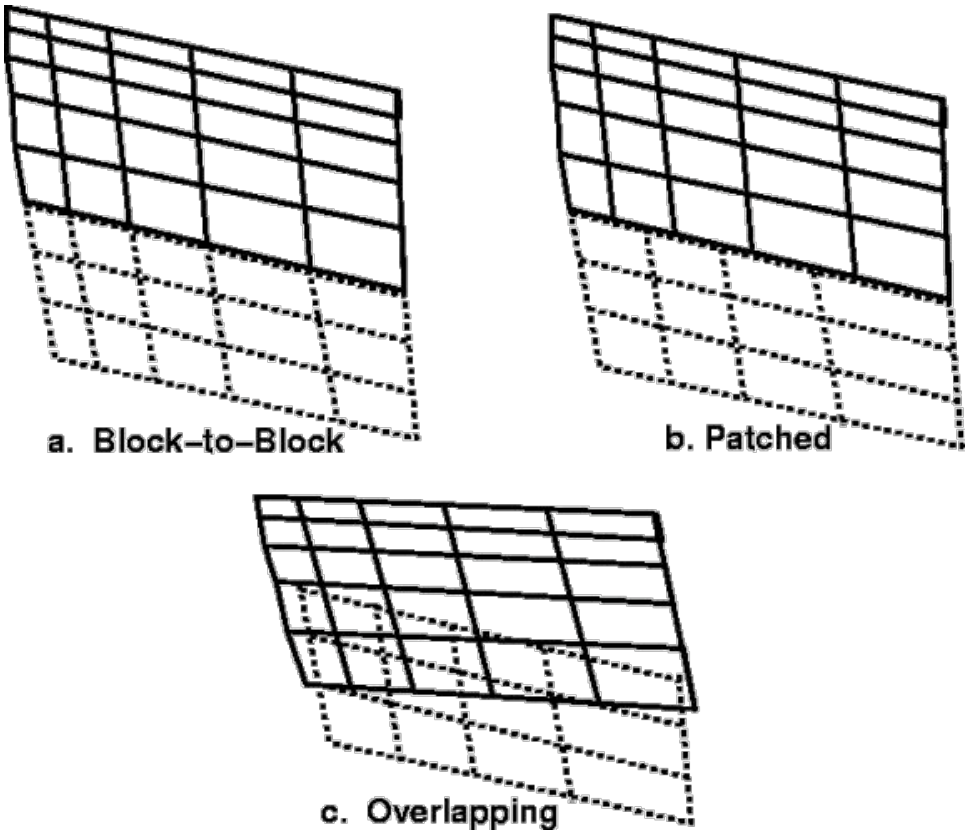


Figure 2: Example of Blocked, Patched, Overlapping Grids

condition is set for the grids, they can then be input into the solver (Beggar in this case) and grid to grid communication is established.

The Beggar code begins this process by taking the inputted grids and initially "cutting holes" for all the solid surfaces, as designated by boundary condition settings. Any cells from an adjacent set of grids are blanked if they lie within another solid. More specifically, Beggar designates a cutting surface and all cells in the overlapping grids as either "in" or "out" of that surface. Each grid that lies "out" of the solid is mapped into all the overlapping grids until the solid has been created in the computational space. Figure 3 shows the fuselage grids overlap with those of the cargo door and thus the fuselage cells that lie "inside" the cargo door have been blanked from the solution process (10). In the subsequent solution process, the fuselage grid cells lying within the cargo door are blanked and thus ignored. This is a very effective method for removing the overlapping cells in a complex solution geometry. After the cells

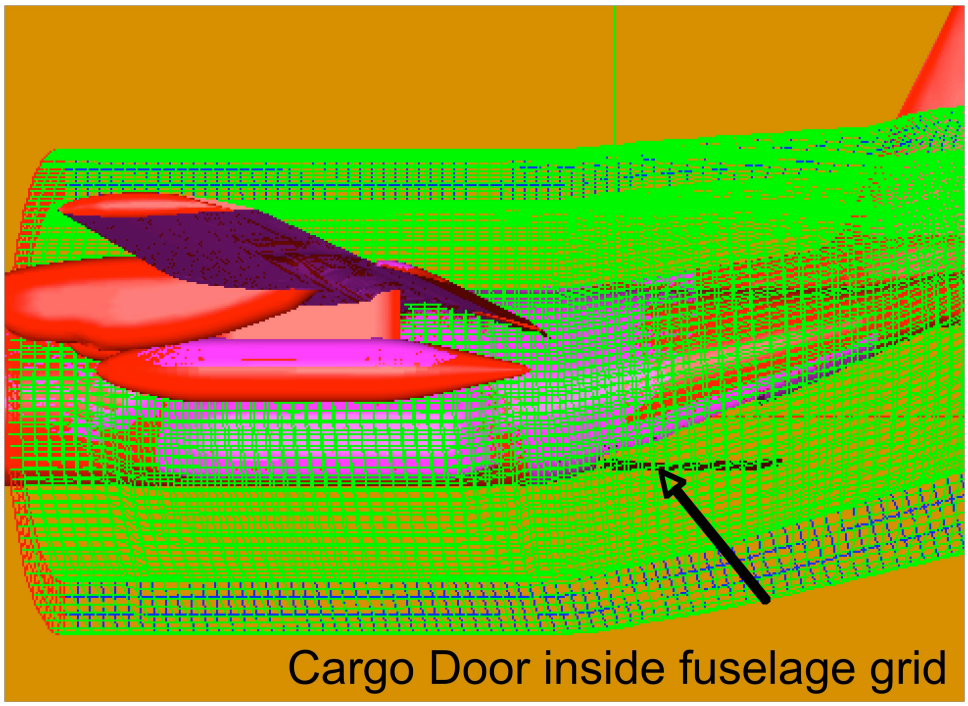


Figure 3: Fuselage and Cargo Door Overlapping Grids

have been blanked, however, there must remain a sufficient number of cells to obtain a valid interpolation of the flow field properties in the region between these adjacent grids. The fuselage grid cells that lie along the declared boundary of the cargo door are on the cutting surface of the cargo door. A minimum of two cells adjacent to the cutting surface, or hole, are necessary for valid interpolation(10). In many cases, the interacting grids do not have sufficient density (or numbers of available cells) outside of the hole when placed in close proximity to have the critical two cells remain for interpolations. In this case, interpolations fail between the adjacent grids and the location is identified by highlighting these "orphan" cells, which are then placed in an output file for the user to examine and fix.

There are two primary options for resolving orphan cells. The first, and more elegant, is to modify the interacting grids, such that they have a greater density in the region of interaction and thus a sufficient number of remaining cells for interpolation. Depending on the stage of solution that these orphans are discovered or the user's timeline to obtain a solution, this approach may be prohibitive. The second option is

to place a dense interface grid in the region where the orphans lie. In this case, the interpolation is solved over this region by the brute force insertion of the interface grid. This "quick and dirty" method was the approach taken to solve interactions between the fuselage, cargo bay door and weapon. Interface grids add cells and increase run times, as well as forcing additional interpolations which leads to decreased overall accuracy.

The ability to have two bodies move relative to one another in a time-accurate, full physics solution is a powerful tool for CFD solutions. As opposed to the solutions outlined in the literature review, the store and the C-130 move relative to one another. However, it is not necessarily obvious where grid interpolation difficulties will occur when the solution is first attempted. In this case, store interactions with the upper surface of the door became a problem as the front of the store passed the trailing edge of the door during the extraction process. Another interface grid was inserted above the trailing edge of the door to correct this problem.

One other significant point should be raised in the overset grids discussion. As is well known, the most accurate solution is obtained in CFD when great care is taken during the initial grid construction. Properties of a "good grid" include uniformity of cell size, orthoganality to the surface, smooth changes throughout the grid and an appropriate number of cells to capture the major flow features. Any time these grids are overset in the cells of an adjacent grid, some of these properties of a good grid are affected. In addition, interpolation always induces error(11). However, the author believes that the ease of solution, and relative motion between bodies far outweigh these disadvantages.

2.3 6DOF integration

Once the grids have been assembled and are communicating properly, Beggar can begin the flow solution. Communication between grids also implies that transformation into the local reference frame of all the super blocks has been accomplished, such that individual points in three space have been mapped into multiple grids dur-

ing the overlapping process and that Beggar will be able to continue to resolve flow values at these points as the solution continues. The output of this process is to provide pressure and viscous stresses along the surfaces of the bodies in the flow field. Integrating these forces along the surfaces of the bodies yields the forces and moments experienced by these same bodies. At this point, the flow solver hands these values off to the 6DOF solver which can then solve the equations of motion for the bodies in question(14). Figure 4 is a flow chart of the process and is included for clarity.

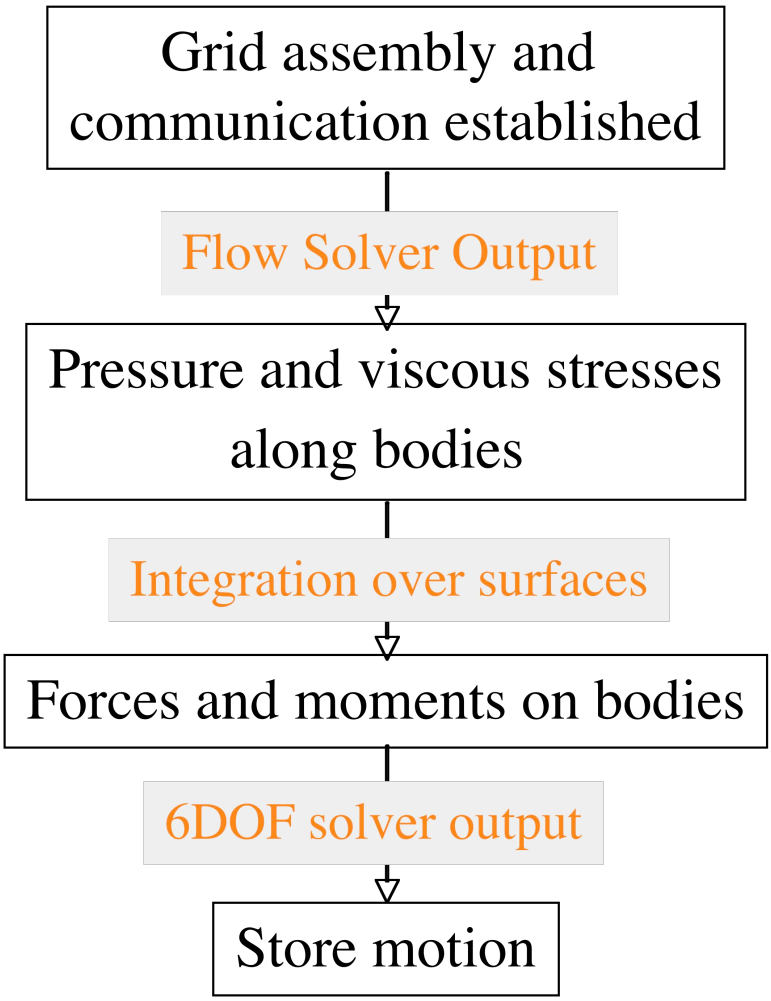


Figure 4: Flow chart of 6DOF solver process

The 6DOF solver has the capability to incorporate moving components on the stores, such as controls surfaces, that would affect the trajectory and orientation of the store in flight. This capability arises from Beggar's ability to "stitch together"

these separate rigid bodies (in the case of the JASSM, the body and its wings) and transfer the aerodynamic loads between the associated grids(14). In this solution, this capability was not used, since no control surfaces were attached to the modeled store. One additional note, Beggar treats all of the bodies in the solution as rigid bodies, and hence there is no allowance made for body flex, torsion, etc.

III. METHODOLOGY

3.1 Grid Generation

The baseline C-130 grids were not suitable for solution as they stood. As stated in the background, they were for an AC-130 gunship and lacked the cargo door or interior bay modeling. In addition, only the left side of the aircraft was modeled as this was the region of interest for the 105 mm howitzer fire. Lastly, the baseline grid lackied any modeling of the propellers on the engines. Based on the fact that the important portion of the analysis was taking place well away from the influence of the engines, there was no attempt made to model the propellers for this project. These grids contained a large number of negative and highly skewed cells. After resolving these issues, the cargo door hole was modeled. From C-130 blueprints it was apparent that the door began at aircraft station number 737 (inches from the nose) and ran to station number 869. Also, the door was given as 120 inches wide by the Systems Program Office (SPO) at Warner-Robins AFB, GA. An x-y plane was created with a width of 60 inches running from stations 737 to 869. This plane was then projected onto the exterior surface of the existing C-130 grid to form the outline of the door. Further in the construction, the entire grid set was "mirrored" to the right side which generated the second half of the cargo door hole, as shown in Figure 5.

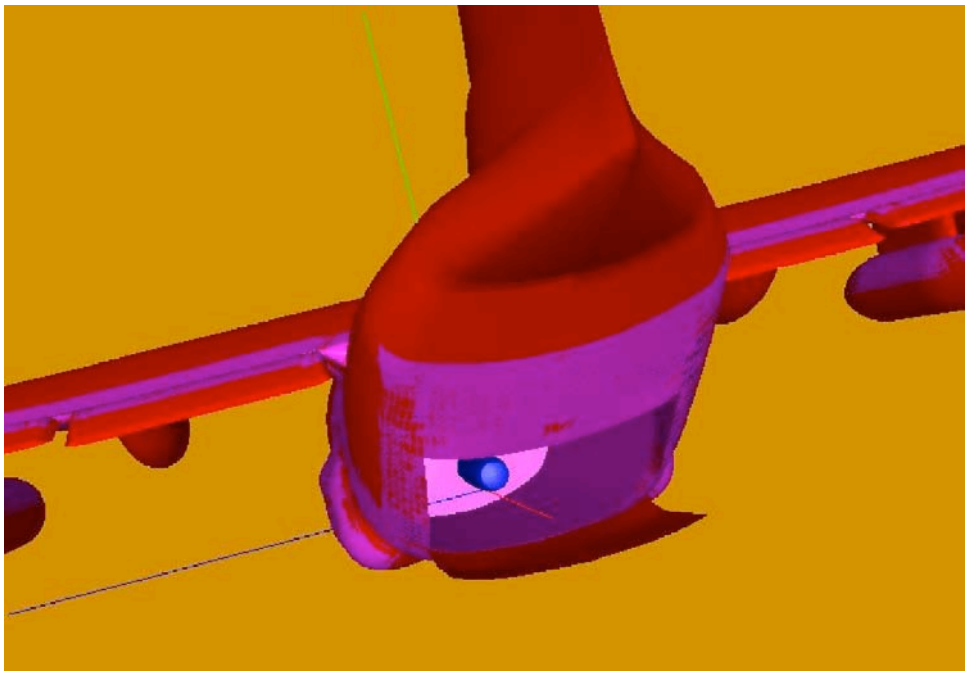
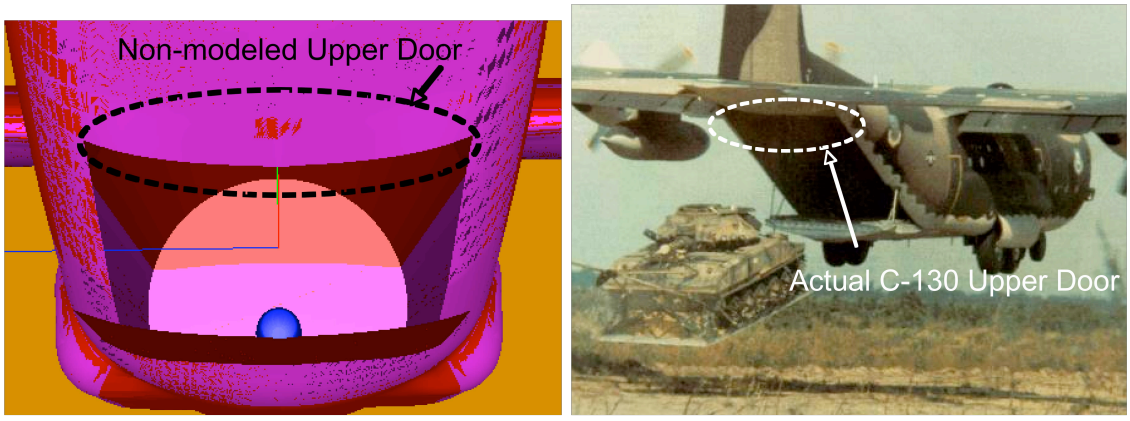


Figure 5: Cargo Bay with Store Inserted in Centerline Position and Cargo Door in Place

Of note for future research, the cargo door itself is actually split into two parts, an upper and a lower section. The lower section of the door is far larger and comprises the entire ramp that is lowered for extractions while airborne or entry and exit while the aircraft is on the ground. The upper portion is much smaller and folds up and away into the cargo bay to provide extra clearance during the cargo loading and unloading on the ground or extraction in flight. This upper portion of the door was not modeled and hence the tolerances of any store that is extracted is marginally tighter and thus requires very careful interface grid construction to obtain a valid solution. The author would suggest adding this upper door cutout to increase the clearance of a variety of store configurations that have been postulated, as shown in Figure 6.

The cargo bay was constructed by scaling the existing fuselage to 95 percent of its current diameter beginning approximately 25 percent back from the nose to the cargo door opening. In this manner, the forward crew compartment space and thickness of the fuselage walls were approximated. This extrusion was accomplished



(a) CFD Non-Modeled Upper Door (b) Actual C-130 Upper Door

Figure 6: Comparison of Non-Modeled Upper Door and Actual Upper Door

by beginning with the existing grid covering the opening that was cut for the cargo door. The grid was "grown" from the door opening cells forward to fill the created void of the cargo bay. The boundary condition for this void was set as an inviscid wall to match the remainder of the aircraft external surface, see Figure 7.

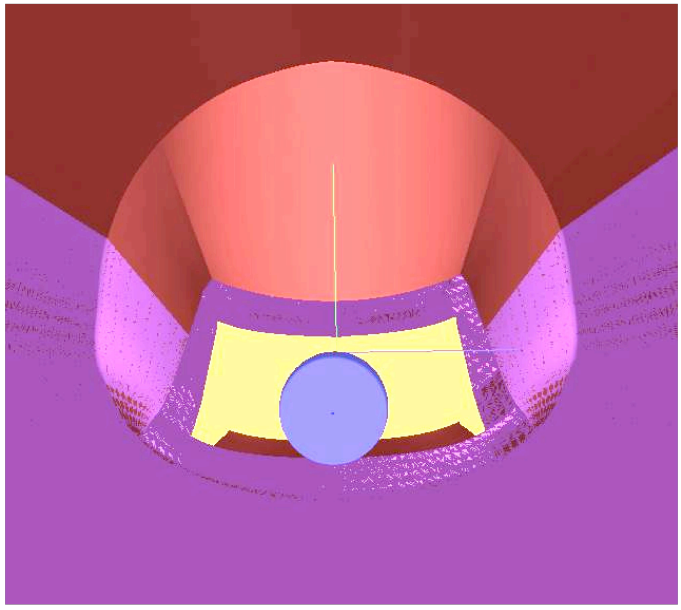


Figure 7: View from Interior of Cargo Bay with Store Inserted in Centerline Position Looking Aft

The cargo door was constructed to match the door hole made in the existing grid. For the purposes of this analysis, the door is essentially two dimensional, in that it has no thickness, as shown in Figure 8. The door was set as a no-flow condition to create a representative flow field in the cargo bay. As opposed the top of the cargo door opening, the author feels that this "thin door" approximation is sufficient for further research and door thickness is not necessary to appropriately capture the major flow field features that would affect stores with the mass and inertial characteristics typical to this analysis. This assumption is based on the fact that a 2,500 lbs store is being released into the wind stream and not something considerably smaller and lighter, like a human body. Johnson et al. (6) were very concerned about the effects of this flow field on paratroopers exiting over the cargo door and hence a further refinement to their cargo door was necessary to capture the thickness of the door. Serrano et al. (15) showed that when the extraction chute was placed in the flow aft of the tail, the airflow was "clean" enough that the effect of the cargo door updraft on the chute was negligible. Based on the C-130 extraction chute data (9), the length of the standard rigging lines are sufficient such that when the chute is fully deployed and beginning to extract the store, the chute should be aft of the region of influence from the cargo door.

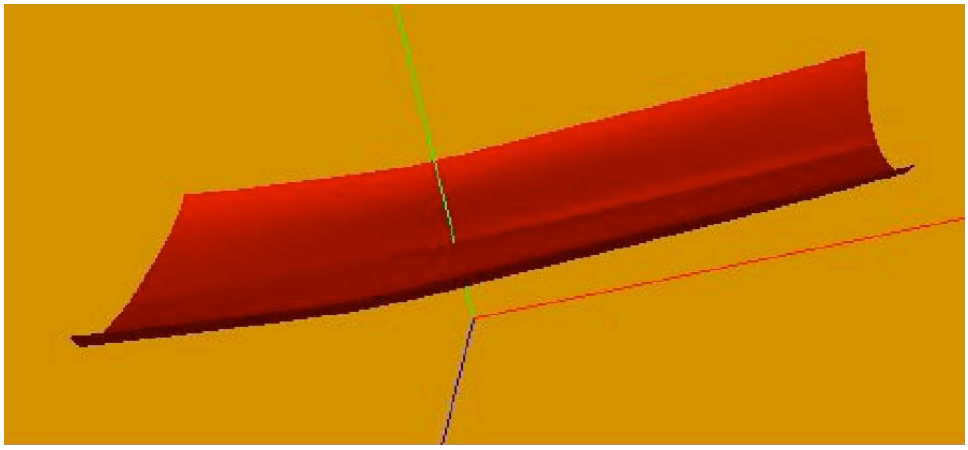


Figure 8: Cargo Door Model

The ideal store grids for this analysis would have been the Lockheed proprietary JASSM grids, but due to issues with distribution of the final product, a generalized store was created. A picture of the actual weapon has been included in Figure 9 for reference. Based on the available data, every attempt was made to match the

Figure 9: JASSM

inertial, mass and scalar properties of the JASSM, but the models developed are far from perfect. For the initial run, a generic "MK-84 like" body was developed. The largest drawback with this model was that its cylindrical shape tended to minimize the influence of the flow field on the store. After the initial run, a vertically "squashed" store was used which more closely models the actual weapon. Of note, the baseline and "squashed" or "flattened" grids were the only grids in the construction that had sufficient cell density to be run with a viscous solution. The parameter y^+ is commonly used to evaluate the suitability of the grid spacing to support a viscous solution and is computed as shown below:

$$y^+ = \frac{y(|\tau_w|/\rho_w)^{\frac{1}{2}}}{\nu_w} \tag{4}$$

Where y is the physical spacing from the wall of the designated cell, τ_w is the shear stress at the wall, ρ_w is the density at the wall and ν_w is the kinematic viscosity, or μ (coefficient of viscosity) over ρ. In Figure 10, a y^+ plot has been attached to illustrate the grid suitability for a viscous solution. Note that the value of y^+ never exceeds 1.2,

where a value of 1 is the typical value aimed for in a Baldwin-Lomax solution (such as Beggar uses), but values under 5 are acceptable. The x-axis on the graph has units of inches from the front of the aircraft and this plot was computed after the store had been extracted and was well into the slip stream to ensure a suitable velocity profile to assess the y^+ parameter(7). Therefore, for the overall solution, the aircraft

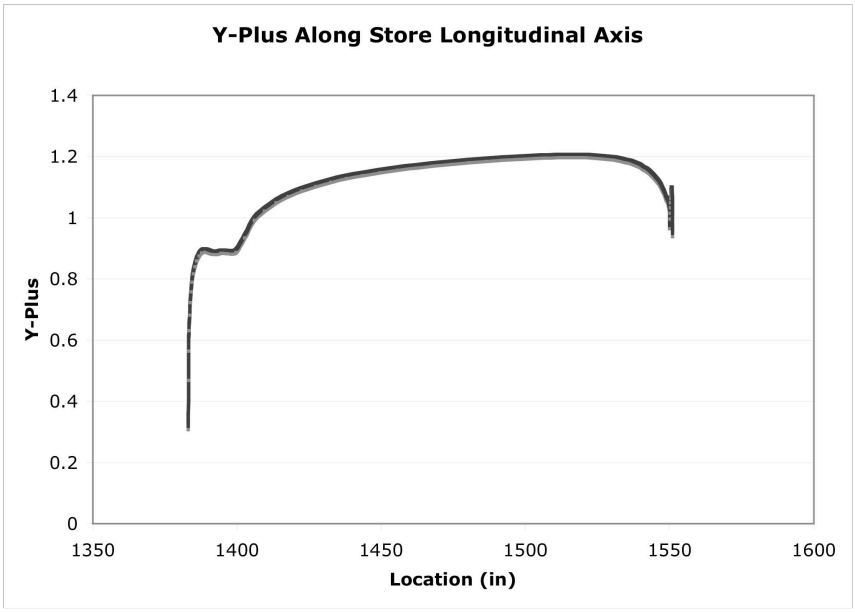
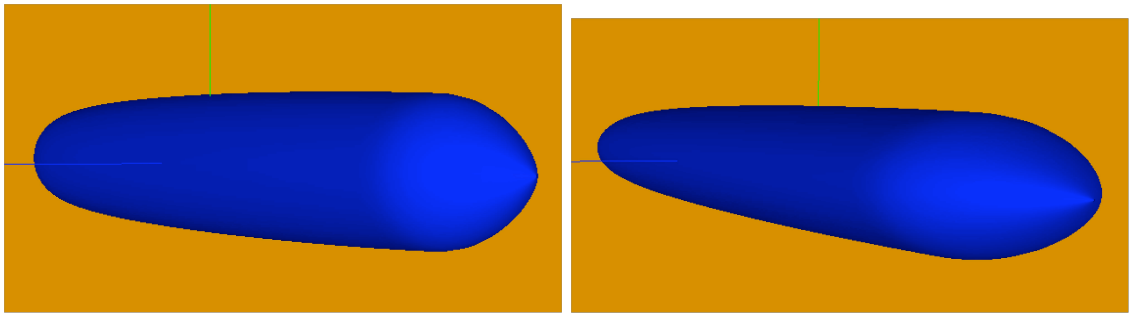


Figure 10: Y-Plus Along the Store Longitudinal Axis

is simulated with a inviscid surface, while the store has a "no slip" viscous boundary condition. In Figure 11, note the comparison between the actual JASSM in Figure 9 and the baseline and flattened stores. Pay particular attention to the"sharp" edges of the JASSM as well as its control surfaces as compared to the grids used in this analysis. Obviously, the stores used in this analysis will minimize the aerodynamic effects of the flow field when compared to the actual JASSM due to their relatively "smooth" profiles. One last note in favor of the validity of the analysis contained herein–the control surfaces of the actual JASSM are not deployed immediately, but slowly extend during the extraction, meaning the flattened store is a reasonable approximation.

The crux of this problem was in the area of the cargo door grids, fuselage grids and weapon grids interface. These three super blocks saw the vast majority



(a) Baseline (b) Flattened

Figure 11: Store Models Used in Computations

of flow field transients and grid interactions during the solution time frame. Given

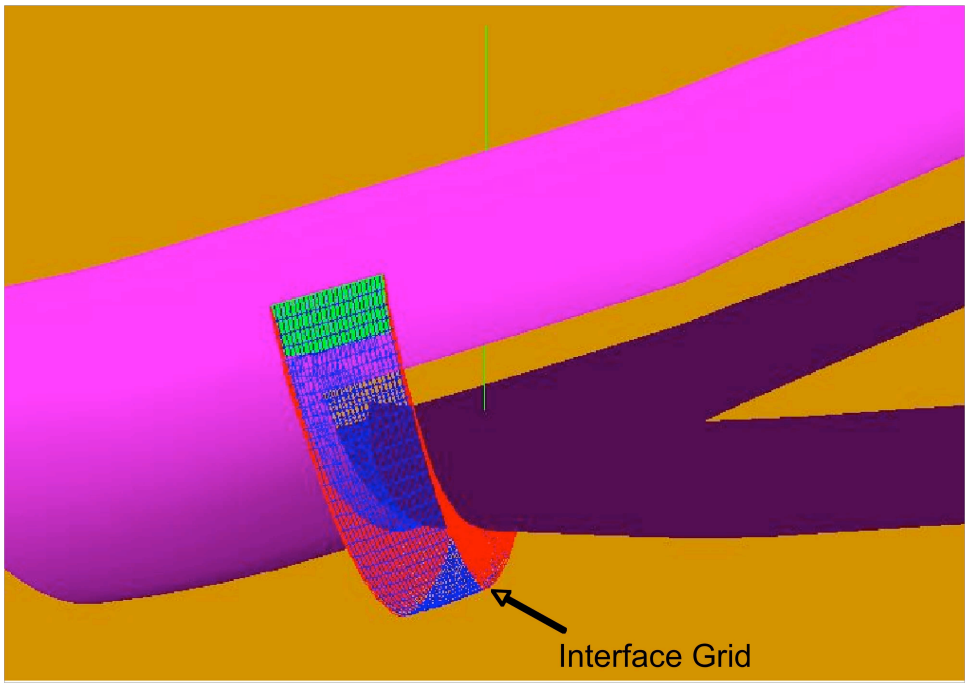
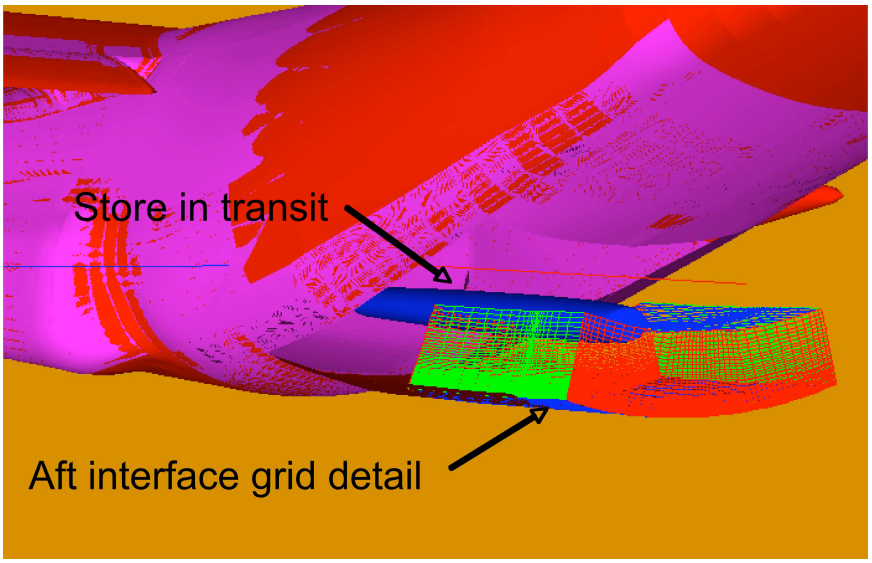


Figure 12: Interface Grid

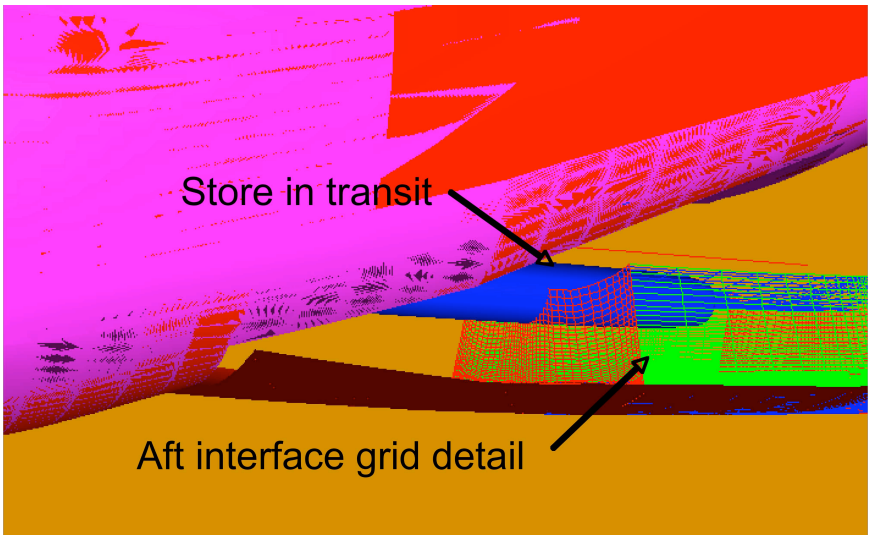
the coarse nature of the overall fuselage grid, from which the cargo door and void were extrapolated, several problems arose. The first problem was on the leading edge of the cargo door. The overset grids had difficulty communicating because of the proximity to the fuselage. As discussed in the overset grids theory section, there must be an adequate number of cells to be "blanked" from overlap and yet still obtain a

solution for the flow field. There was only 1 cell between the door and the fuselage originally. After the problem was isolated to the region shown in Figure 12, two additional interface grids were constructed and placed over the region of interest.

The other region of difficulty was over the top surface of the door and particularly over the trailing end as the bomb travelled over this area during extraction. The store, which had a dense grid to support the viscous boundary condition, had to maintain valid interpolations with the less dense door upper surface. Specifically, as the store reached the end of the door and began to fall and pitch slightly, there was a region of failed interpolation. Once again, the solution was to place one more interface grid on the trailing edge of the upper door surface, as shown in Figure 13.



(a) Rear Quarter View



(b) Lateral View

Figure 13: Cargo Door, Store Interface Grid Detail

3.2 Grid Dimensions

After all was completed and assembled, the final series of 188 grids contained just over 2.5 million cells, comprising a total of 21 super blocks. The store itself, contains, over 200,000 cells with a y^+ranging between 1 and 1.2 and the initial spacing of 0.01 inches off the wall. Utilizing the parallel processing capabilities of Beggar and running on an average of 16 nodes of single processor, 1 GB RAM AMD Athlon XP 3000+ machines at the Air Force Institute of Technology (AFIT), the average job took 3-4 days from start to finish.

3.3 Parachute Extraction Force Simulation

The CAT shop in the AFSEO is close to a full-up simulation of a parachute with time accurate forces. However, that model contains some 10 million points, and is not yet ready to be integrated into this problem. In addition, solution time would have been prohibitive with that many cells; therefore a fairly gross approximation was made for the parachute force acting on the store during extraction. The data for this analysis was obtained from the Air Transportability Test Loading Agency division of the Air Systems Command/ENFC division in their yet to be published C-130 Airdrop document which covers extraction configurations and experimental force loads(9). Table 1 provides the force for a 140 knot delivery with a 15 foot chute. The baseline run extraction force used for this analysis was 6,150 lbf.

Table 1: Extraction Chute Towed Force

Speed (KIAS)	15-FT RS	22-FT RS	28-FT RS	2 28-FT RS	3 28-FT RS	35-FT RS	2 35-FT RS
			C-130				
110	3,780	6,170	16,700	30,428	43,261	29,550	50,700
120	4,510	9,760	19,870	36,213	51,485	35,170	60,400
130	5,340	11,500	24,200	42,500	60,424	41,270	70,900
140	6,150	13,600	27,050	49,289	70,070	47,870	82,200
150	7,050	15,200	30,000	56,580	80,446	55,000	94,350
160	8,050	17,400	35,300	64,378	91,529	62,600	107,350

This 6,150 lbf is a peak force obtained by the chute during extraction and was the starting point used for the initial runs during this analysis. This peak force is

higher than what would be transmitted to the store over the entire extraction due to the change in shape of the chute as it expands and contracts in deployment, then slows down in the free stream. Initially, though, the concern was whether the store would clear the aircraft prior to dropping on the floor and ceasing interpolations. For the baseline run the peak parachute force was applied to extract the store as quickly as possible. A constraint was set to hold the store in place as it traveled down the cargo bay and prior to exit into the wind stream. This constraint simulates having the store on a sled necessary for parachute extraction. The constrain was in place for 0.8 seconds and inhibited vertical and lateral movement but allowed fore-aft movement (x-direction). This constraint time was developed after a "trial and error" series of runs to determine the time elapsed from the store release at the front of the cargo hold to the threshold of the door (the time the store would normally be released from its carriage sled, since the store would obviously not float 15 inches off the floor down the length of the cargo bay of its own accord). Another simplification was made in which the parachute force was applied directly to the center of gravity of the store and along the store centered x-axis, such that only aerodynamic loading would cause the store to change its orientation. This simplification once again approximates sled carriage of the store(7). Under these conditions, regardless of the store orientation, the parachute force continues to act along the longitudinal axis of the weapon.

Once the "proof or concept" runs had been accomplished a more realistic parachute model was incorporated. This model more accurately represents the time history force profile of a typical parachute as it "blossoms", captures the wind stream, then begins to slow during deployment. A typical generic force profile curve has a shape as shown in Figure 14(8):

Note that an extracted store will normally rest on a palette that is held in place with floor locks that are calibrated to release at a specified force. According to ASC/ENFC division at Wright-Patterson AFB (WPAFB) a reasonable force for release of the locks is roughly 3,500 lbf. At this point the store will begin to see the force of the chute and begin to move. In under a tenth of a second, the parachute

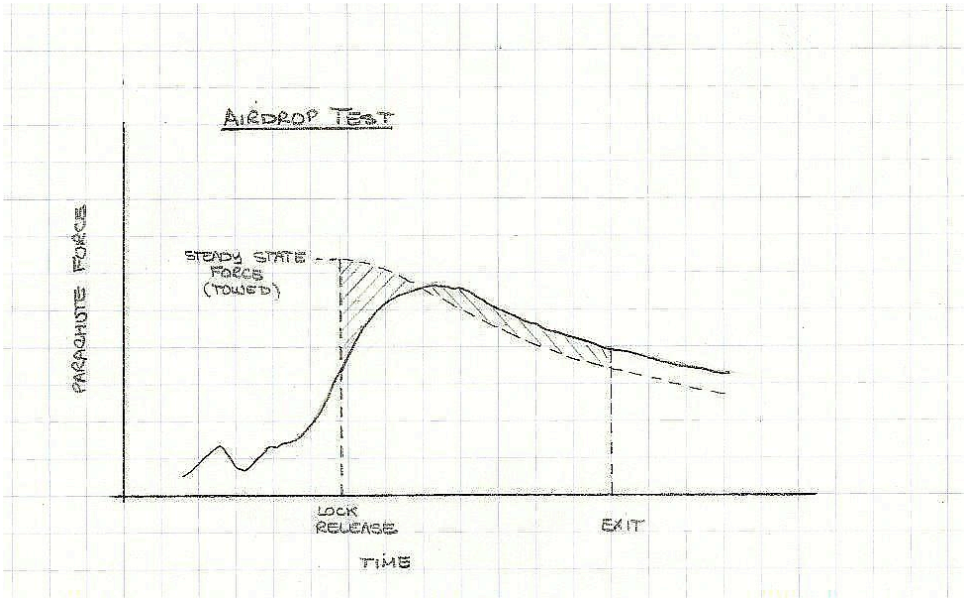


Figure 14: Extraction Force Generic Profile

has fully blossomed, reached its peak and then is beginning the force "falloff". This "falloff" is the result of the reduction of dynamic pressure on the chute as it slows in the wind stream. The relationship used for the majority of this analysis represents the peak and subsequent falloff of the chute in a time accurate model coupled with the CFD code. The relationship utilized is shown below (8):

$$F(t) = \frac{1}{2}\rho \left(\frac{1}{\frac{1}{2}\rho C_D S_o (\frac{g}{w})t + \frac{1}{V_p}} \right)^2 C_D S_o \tag{5}$$

In which, ρ is density, $C_D S_o$ is given in the Airdrop data folder as 97 (8), g is standard gravity in English units, w is weight of the store, given as 2,500 lbf and V_p was assumed to be the release speed of the parachute for the given extraction conditions of 140 knots, or 236.6 ft/sec. Given these conditions, at 0.01 sec into the drop, the extraction force the store sees is 6,411 lbf while 1.5 sec into the run the force has dropped to only 2,768 lbf. Figure 15 below shows a graph of this force as a function of time. Note that this function does not model the initial force rise shown in Figure 14, only the "force fall-off".

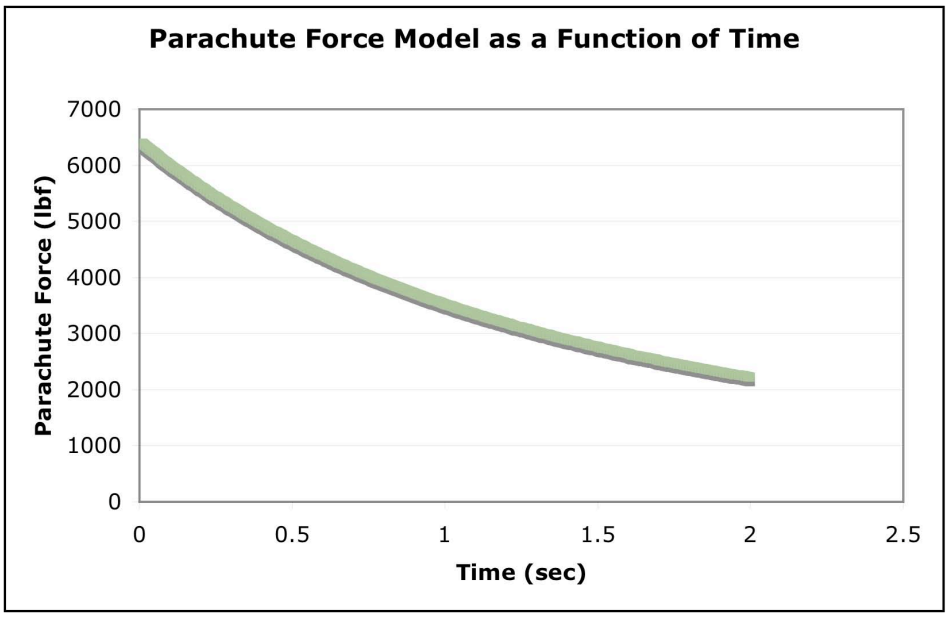


Figure 15: Parachute Force Model as a Function of Time

3.4 Beggar Inputs

Beggar has a wide range of input variable and run-time controls that allow the user to modify the solution both before execution and during iterations. The complete input file has been attached in Appendix A as well as the Beggar input files for the individual runs that modified the store location, AOA and parachute force in Appendix B.

3.4.1 Input File. The majority of the input file is calling the various grids and ensuring the super blocks are correctly assembled for the follow-on solution, but particular attention needs to be placed on the setup parameters used to obtain the runs. In this section, the user specifies all of the run conditions, the solvers to be used for solution and their parameters. The AFSEO CAT has a standard set of inputs for the class of problems they most commonly run and these inputs were the basis for the runs contained herein(7). These settings arrive at an "engineering" solution while avoiding excessive run times commensurate with the USAF's operational need

for a large number of computed store carriage and separation solutions. A line-by-line discussion of the input file is listed in Appendix A.

3.4.2 Beggar Grid Input Files. All the input files for the various grids have been placed in Appendix B with an explanation of their syntax for reference. In general, these files are clear text inputs in which the grids are read in and subsequently translated or rotated as necessary to mate with the remainder of the grids for assembly. The dynamic input files have also been included which are used to establish inertial properties, force specifications and location of force application. Together these files establish the boundary conditions for all surfaces (e.g. inviscid wall, viscous wall, velocities at the wall, etc.) as well as any scaling and relative location of the actual grids used during the solution. It is through these input files that store was placed in different orientations and "flattened" to more closely represent the actual JASSM.

3.5 Selected Runs

The initial run was accomplished with the generic store aligned to the centerline of the cargo bay with the peak force of the parachute applied to the store. After that, the lateral constraints available that allowed for valid interpolations throughout the run were explored. With the store offset 30 inches to the side, valid interpolations were possible. Having established the limits available, the "flattened" store which would more accurately represent the shape of the JASSM was incorporated on this run and all subsequent runs. The "flattened" store was run down the centerline and in the fully offset position. Next, the time varying model for the parachute force was instituted. After running one more centerline run with the time varying model, the angle of attack was increased, based once again on the Airdrop historical data (12), to 5 degrees. Both the centerline case and the offset case were rerun. The results will be presented for these runs in the next section. Table 2 below summarizes the runs completed for this analysis:

Table 2: Summary of Runs Analyzed

Run	Store Type	Postion	Parachute Model	AOA (degrees)
1	Baseline	Centerline	Peak Force	0
2	Flattened	Centerline	Peak Force	0
3	Flattened	30 inches Offset	Peak Force	0
4	Flattened	Centerline	Time Accurate	0
5	Flattened	Centerline	Time Accurate	5
6	Flattened	30 inches Offset	Time Accurate	5

All runs were accomplished at the following conditions: Mach = 0.22, Re/in =1140.0 (a dimensional Reynolds number varying per inch since, in these problems, there isn't a standard reference length), CFL = 30,000 and standard day conditions at 5,000 MSL. The 5 degree AOA runs modified the velocity vector of the flow stream by rotating a negative 5 degrees and the gravity vector was then split into two components of 2.8 in the x-axis and -32.1 in the y-axis.

IV. RESULTS

4.1 Baseline

Prior to any full motion runs it was necessary to initialize the flow field and ensure that the major flow field features had settled out. The field was initialized with the free stream conditions and a non-dimensional force specification tracker was placed on the cargo door to watch for the transients to settle out. Once they had come to rest, the store could be allowed to move. Figure 16 below contains a plot of one of the door force plots to illustrate the point:

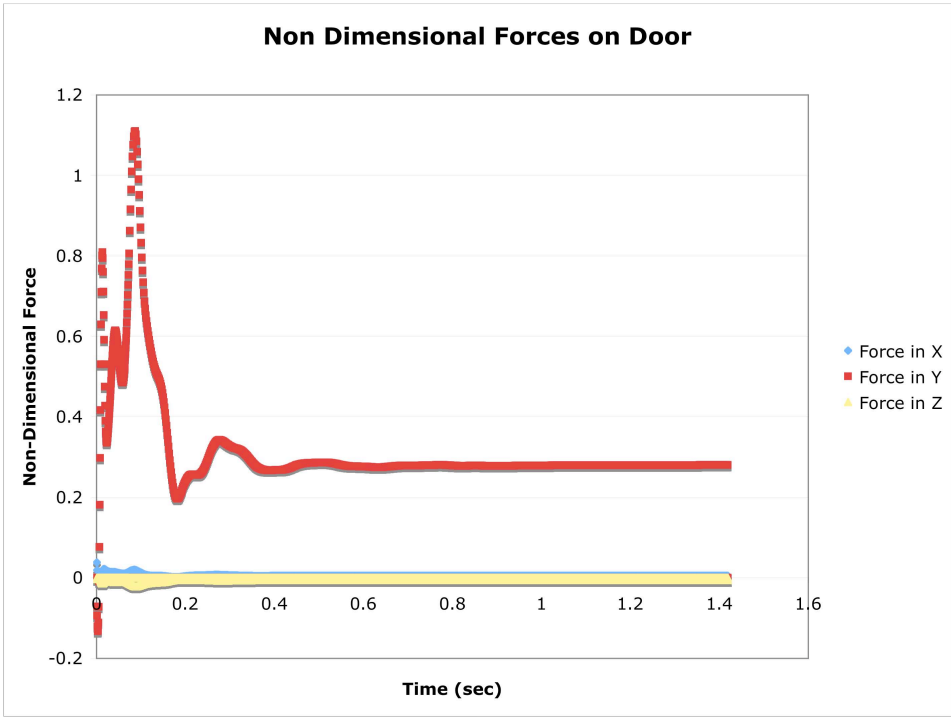


Figure 16: Non-Dimensional Door Force as a Function of Time

Note that after only 0.5 sec, the forces have all but settled out on the door, however, the run was continued out to 1.4 sec (or 1,000 iterations) for completeness. This was the starting point for all store extraction runs presented below.

4.1.1 Centerline Release. As discussed above, the baseline run was a proof of concept that allowed grid interpolation and force specification issues to be resolved.

This run was meant to be to easiest to complete and was placed in the heart of the release envelope to ensure success. Figure 17 illustrates the placement of the store in the cargo bay looking aft.

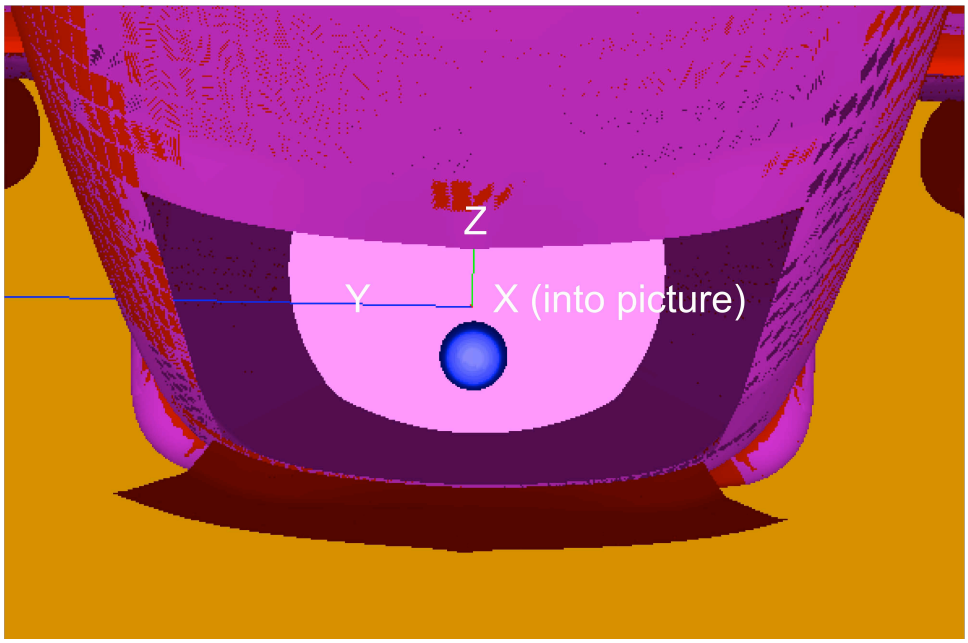
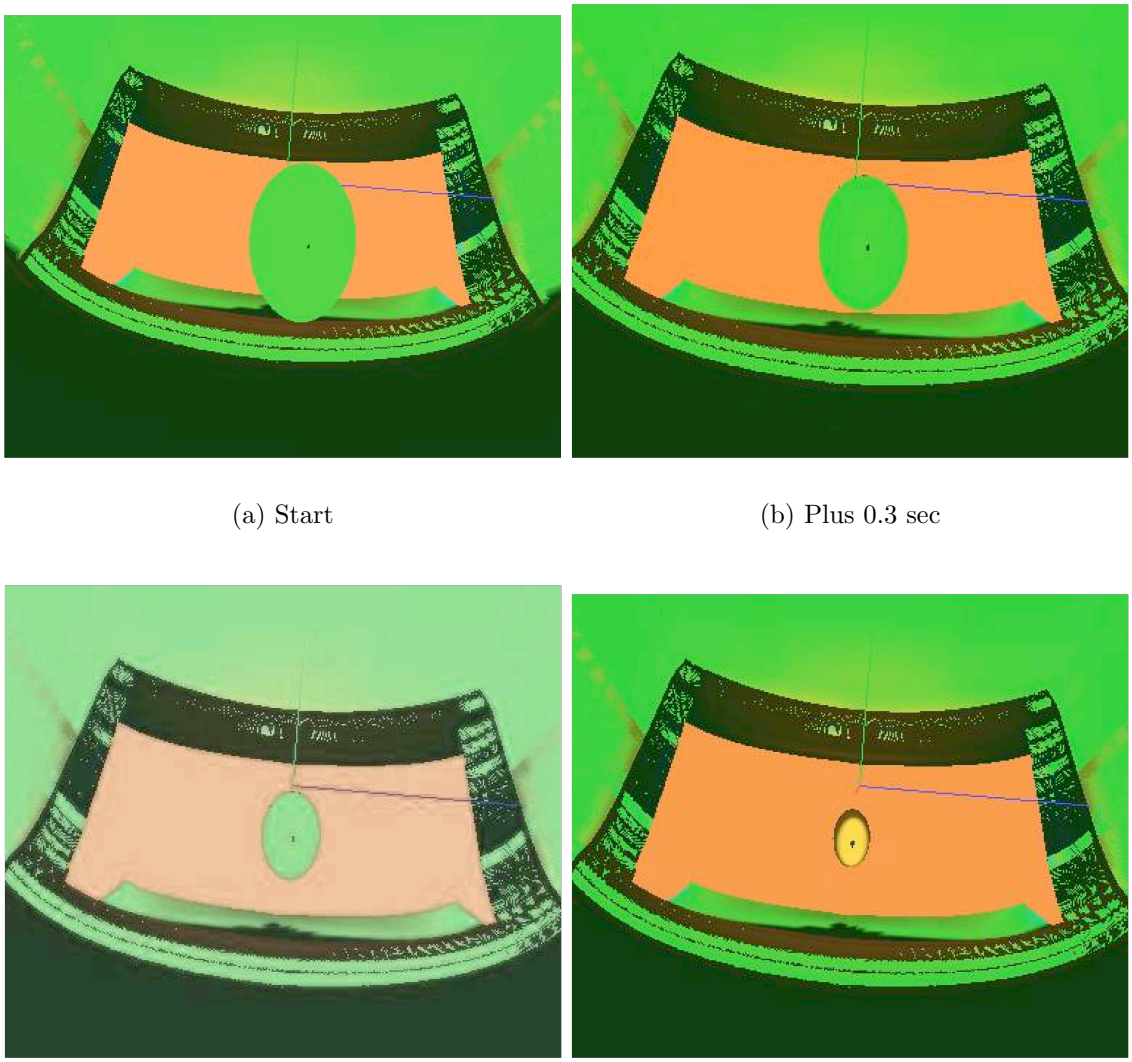


Figure 17: Store in Center of Bay

Given the constant force and immediate application of that force, it is not surprising that the store cleared the aircraft and showed very little motion other than a gravity influenced extraction arc. What follows are a series of plots of the store during the extraction to give the reader a feel for the motion being discussed. Figure 18 shows the store on the way out of the cargo bay as viewed from the interior of the cargo bay.

Figure 19 shows actual position as a function of time as well as angular motion relative to the store initial orientation (remember that in the standard coordinate system, z is down and x is forward).

The store shows a very slight tendency to pitch down then back up as it transits the end of the door ramp (the 40 foot point on the plots). This can be explained by a



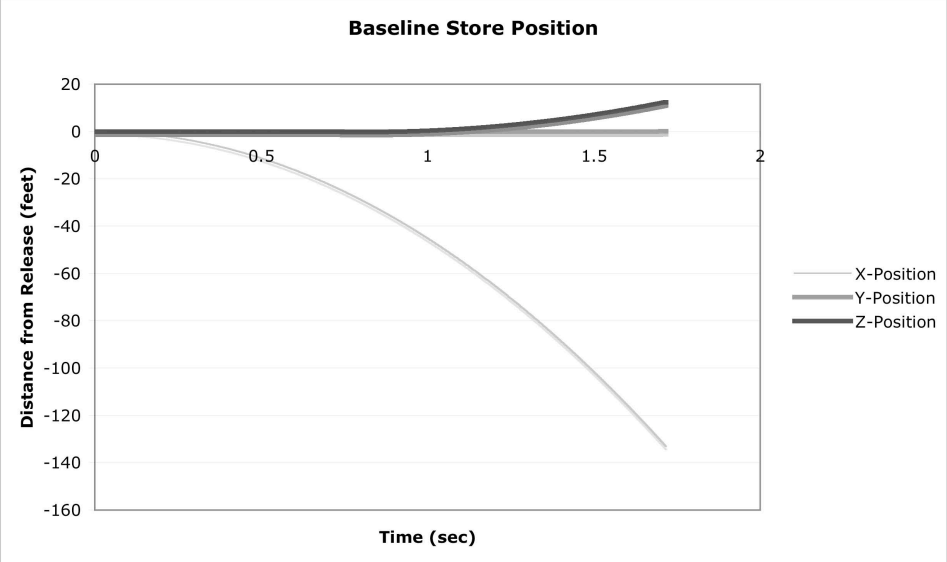
(a) Start

(b) Plus 0.3 sec

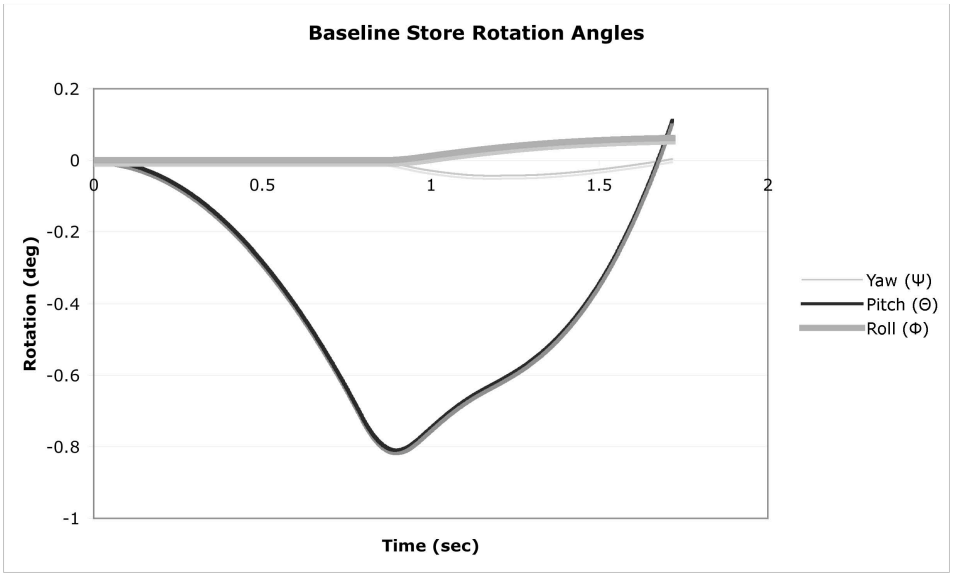
(c) Plus 0.6 sec

(d) Plus 0.9 sec

Figure 18: Baseline Store Exit Visualization as Viewed from Interior of Cargo Bay Looking Aft



(a) Position



(b) Angular Displacement

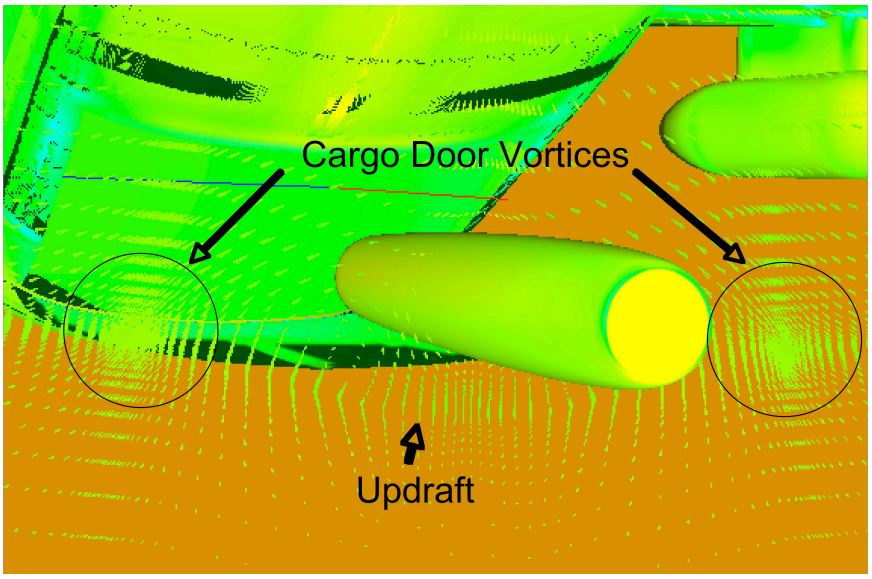
Figure 19: Position and Angular Displacement of Baseline Store as a Function of Time

combination of Beggar solver constraints and an examination of the flow field in this region.

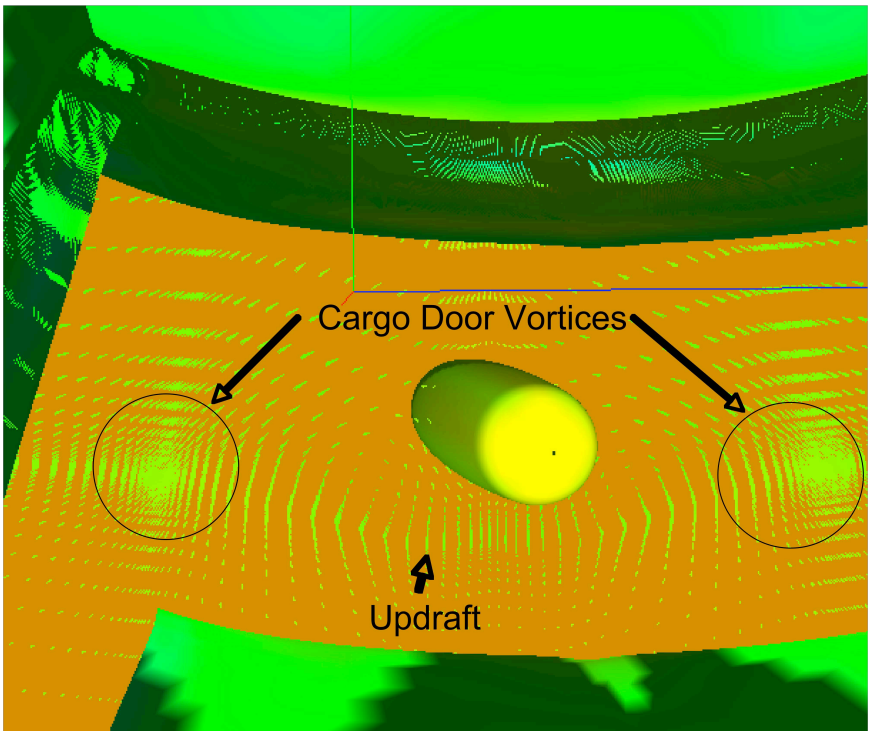
First, when Beggar uses a constraint on movement, as in the case of the store motion down the length of the cargo hold prior to exit, it is not truly applied in a straight line. Based on the way in which the code is written, the constraint must be applied along an arc. In this case, the arc was set in the store motion dynamic specification file to the limiting case of arc radius (i.e. longest radius and lowest curvature). In this manner, a straight line constraint for store motion could be approximated. However, there is a very slight arc on the constraint path which causes the very slight pitching motion seen in all cases prior to exit from the cargo bay. Once the constraint is released at the forward edge of the cargo door (approximately 40 feet into the extraction along the x-axis), the store is allowed to move according to physics and the aerodynamics forces begin to effect the store's motion. Note that generally, there is some energized flow that curls up and over from the bottom of the door to the top along the trailing edge, which is to be expected, as shown in Figure 20. As the store transits this region there is a very a slight pitch up. The high energy flow circulating from the underside of the door upward produces an updraft that acts along the length of the store as it transits the region shown. This updraft causes a reversal of the constraint induced pitch as seen in the graphs in Figure 19. As would be expected with a circular store, there is almost no roll or yaw during the extraction. Also, note that the scale on all the rotation plots has been expanded to show the motion of the store. In the later runs there is an increase in motion that the reader may miss if the change in scale is ignored.

4.2 Flattened Store

4.2.1 Centerline Release. Once the baseline run had been accomplished, the flattened store was utilized to more closely model the intended weapon. Initially, the flattened store was extracted through the middle of the cargo bay with the full



(a) Rear View



(b) Front View

Figure 20: Flow Field as Baseline Centerline Store Transits Region

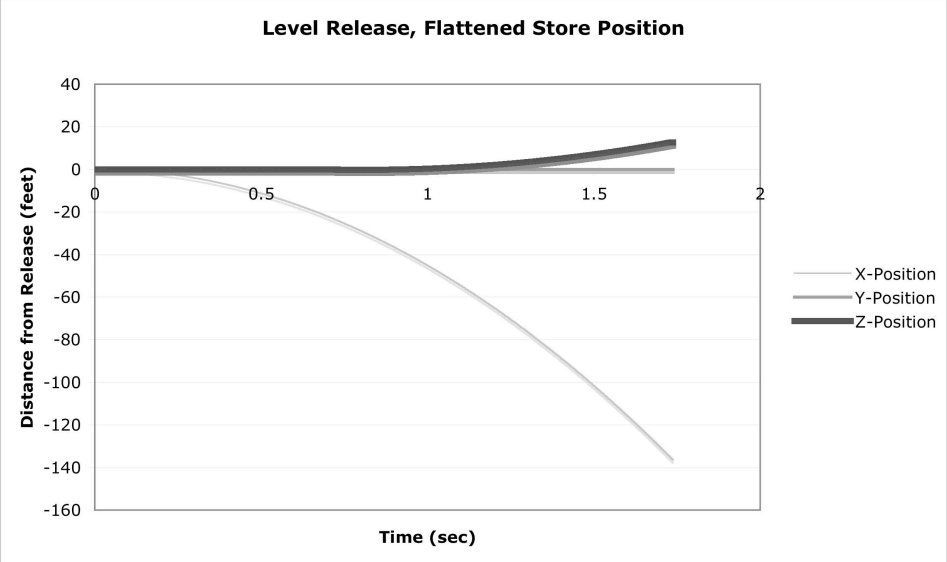
extraction force applied at the CG. The plots for this centerline run are shown in Figure 21.

Note the distinct difference in the roll on the store as it transits the region of the trailing edge of the cargo door. While the influence of the aerodynamic loads is small given that the store has rolled less than a degree at the end of the run, there is clearly an effect. The vortices that curl in from both door edges are impacting the flattened store on the edges and creating a small rolling moment shown in the rotation plot in Figure 21. The author expects this force to be more pronounced when the actual store is integrated into the solution. One important note is that in neither case is there a tendency for the store to rise toward the empennage during the extraction (remembering that z is positive downward). Specifically, the delivery aircraft could be severely damaged by re-contact with the store if the store were to begin a violent pitch up and climb upon exit from the cargo bay due to the updraft.

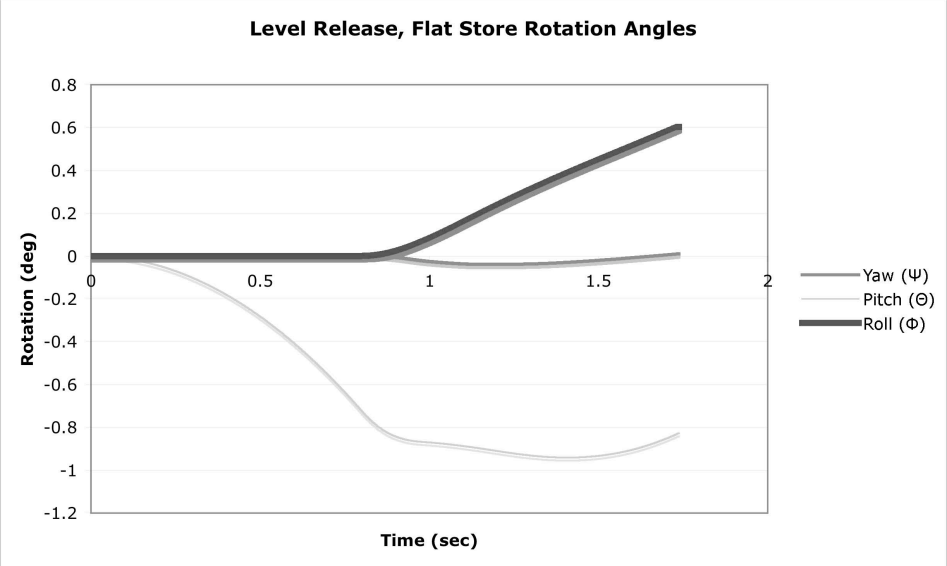
4.2.2 Offset Level Release. Prior to an application of the time accurate parachute force model, one more run was accomplished with the full extraction force to ensure the store would clear the aircraft over the vortices along the lateral edges of the cargo door. Once again, a series of snapshots of the offset store on the way out of the cargo bay are included in Figure 22 to improved the reader's visualization of the process.

The bottom line for this run is that, while the effect of the aerodynamic loading was more pronounced, the overall extraction procedure proved to be safe in this condition. This run was accomplished with the store offset 30 inches from the centerline of the aircraft (Figure 23) through the more "energized" flow along the cargo door edge.

An examination of the velocity vector profile for this run as the store passes the trailing edge of the cargo door shows the more pronounced vortices circulating (in a counter-clockwise direction as viewed from the front) around the outboard edge causing the increased store rotation and pitching motion (Figure 24). The aerody-

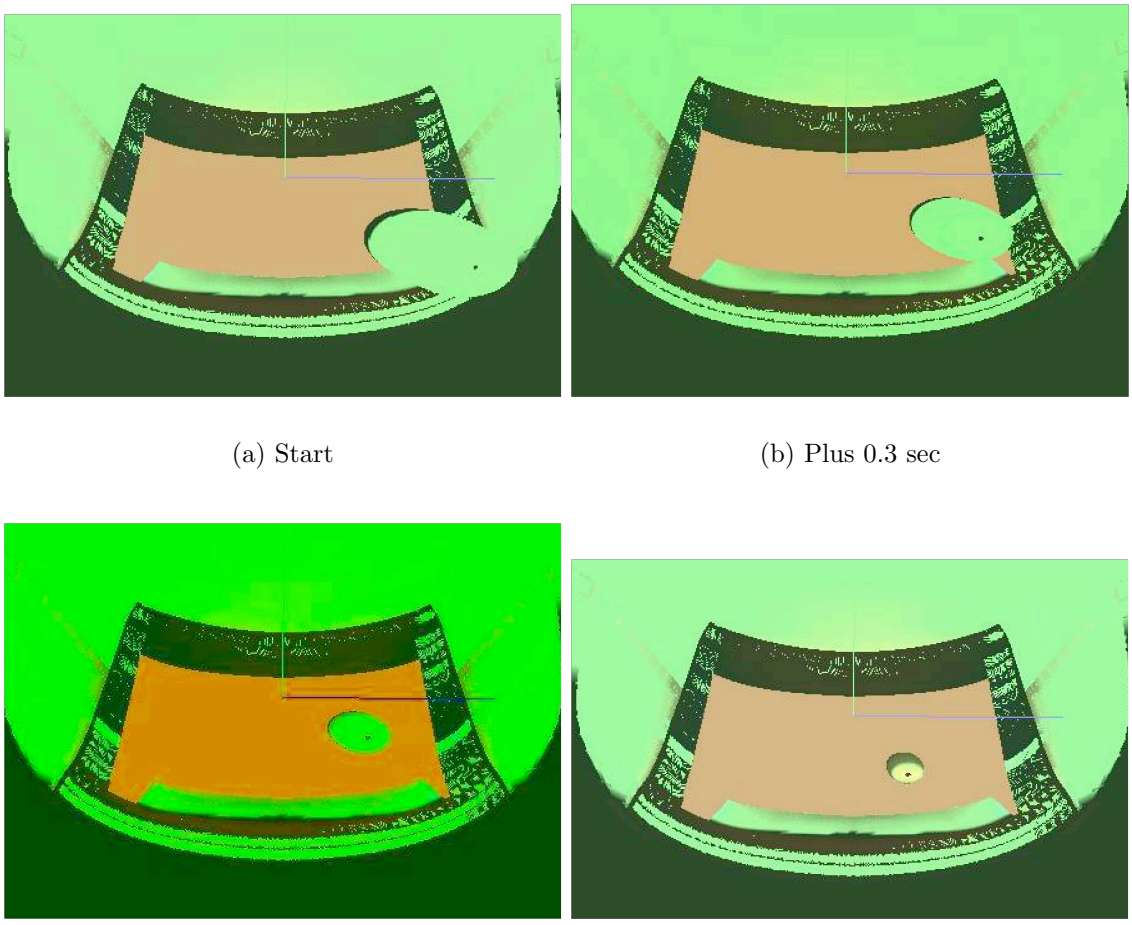


(a) Position



(b) Angular Displacement

Figure 21: Position and Angular Displacement of Flattened Store as a Function of Time



(a) Start

(b) Plus 0.3 sec

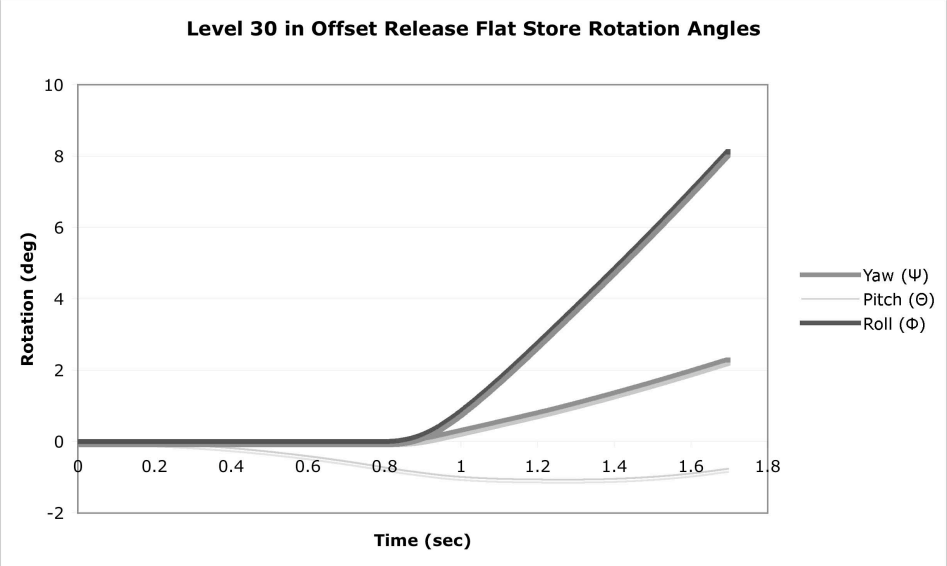
(c) Plus 0.6 sec

(d) Plus 0.9 sec

Figure 22: Offset Flattened Store Exit Visualization from Interior of Cargo Bay Looking Aft

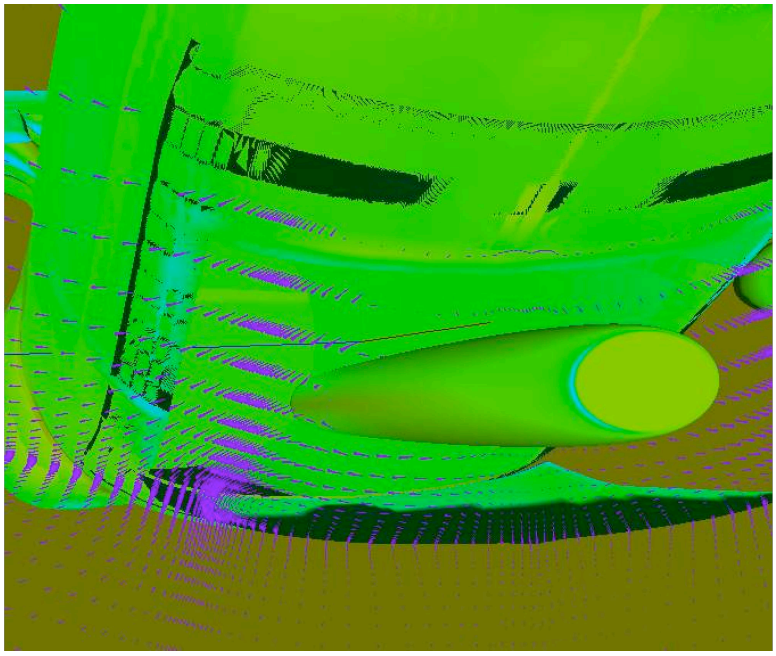
namic loads acting on the store in this region are significantly larger than those down the centerline of the cargo door and it would be worrisome if the store did not show greater rotations during these "offset" extractions.

(a) Position

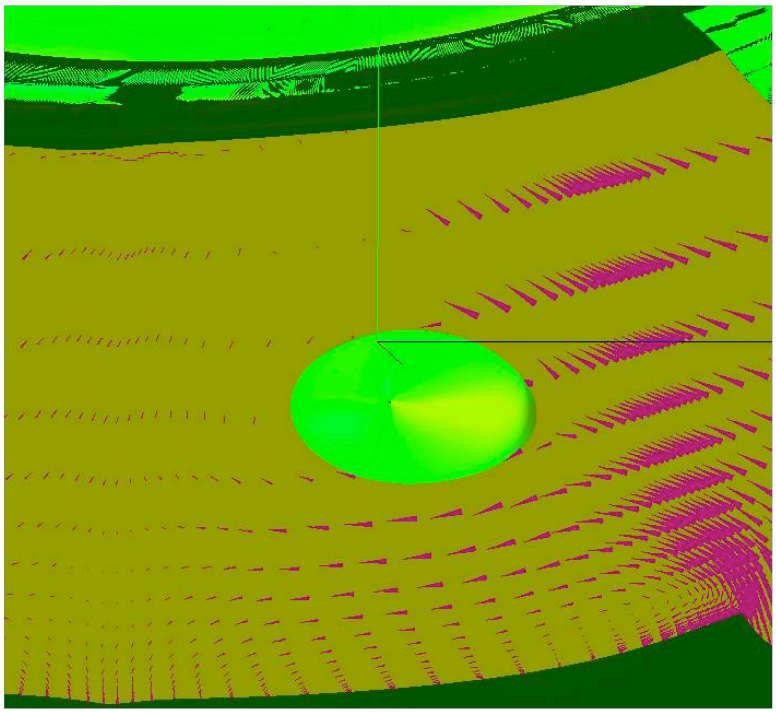


(b) Angular Displacement

Figure 23: Position and Angular Displacement of 30 in Offset Flattened Store as a Function of Time



(a) Rear View



(b) Front View

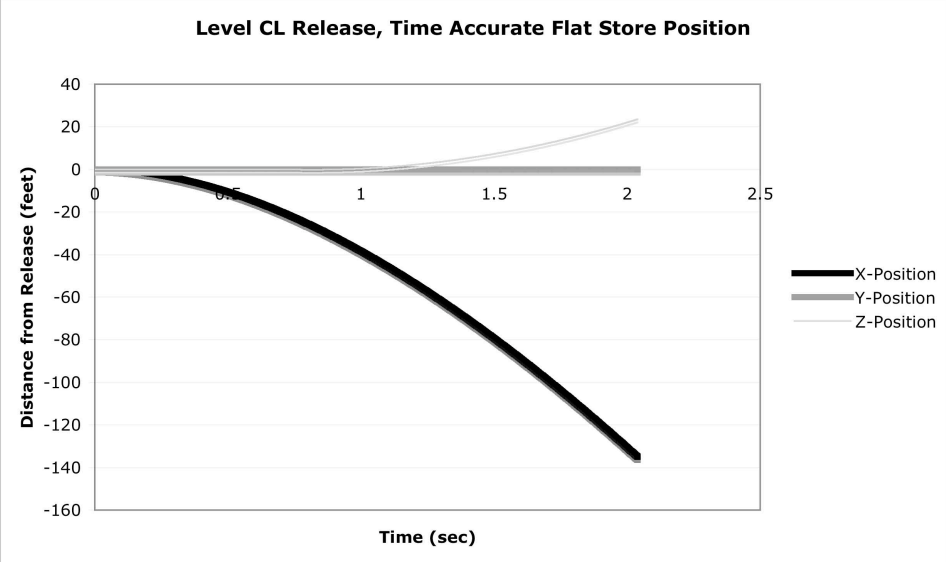
Figure 24: Flow Field as Offset Flattened Store Transits Region

4.3 Time Varying Parachute Extraction Force

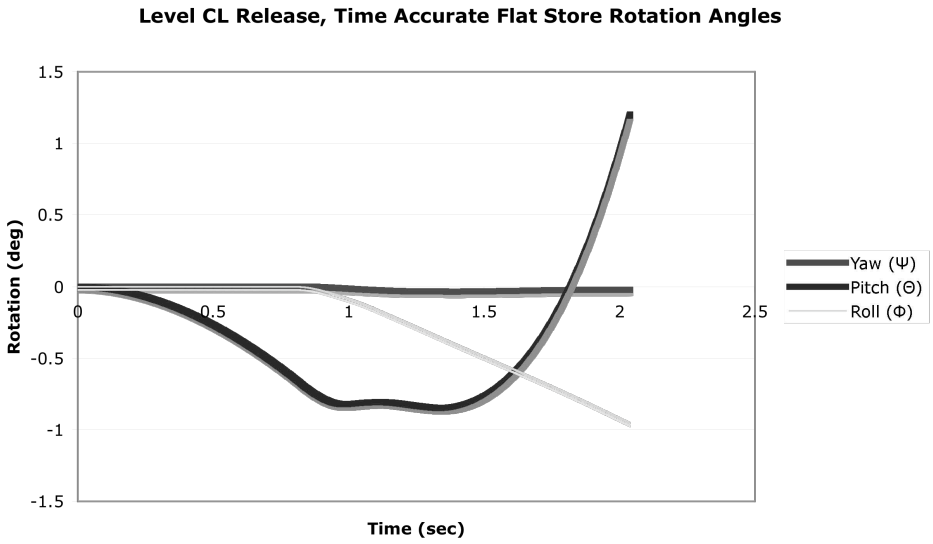
4.3.1 Centerline Level Release. The first time varying parachute extraction force run accomplished had the flattened store placed in the center of the cargo bay and extracted with the "peak to falloff" model discussed in chapter 3. The overall effect was to see that the extraction timeline lengthened, but the major features of the procedure remained constant. As you can see in the figures below, the overall time until the store has reached the edge of the "drop box" and interpolations cease increases. The decreasing parachute force as the store is extracted leads to the longer timeline to reach the edge of the far field grid where the solution stops. Most importantly, though, even with the reduced force, the store still has no tendency to climb into the empennage, see Figure 25.

4.3.2 5 deg AOA Centerline Release. The next run was identical to the previous run with the exception of 5 degrees AOA on the aircraft; a more representative flight condition during actual airdrops. As expected, the increase in AOA on the aircraft increases the strength of the vortices generated by the cargo door. Note the increase in both the pitch and roll on the store due to the stronger flow field updraft and circulation aft of the door, and yet, still the store travels a safe arc away from the aircraft as shown in Figure 26.

4.3.3 5 deg AOA 30 in Offset Release. The final run was accomplished with the store set in the offset position and using the time varying parachute model. Of all the extractions seen, this case showed the largest roll which approached 13 degrees (note scale) as the store transits the highly energized vortices along the cargo door edge (Figure 27). Of the runs completed, this extraction simulation most closely represents the case of a store loaded on a sled in a cargo bay full of many other weapons. In this case, the store could be the first of many to be extracted if the weapons were arranged laterally across the cargo bay. Obviously, it would be critical to ensure the first store clears the bay safely and does not foul the exit corridor for subsequent extractions. From a macro-perspective of the store exit visualization, there is not a great deal of

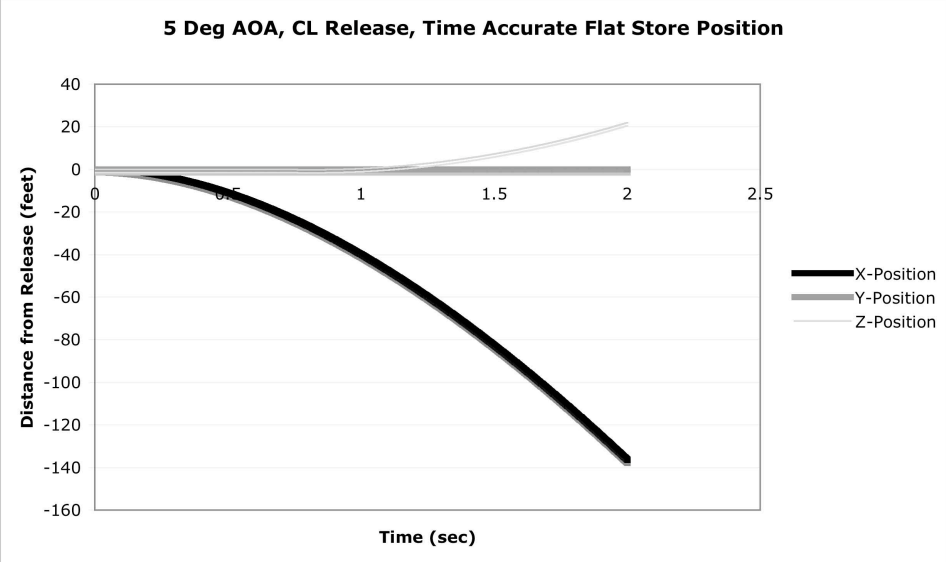


(a) Position

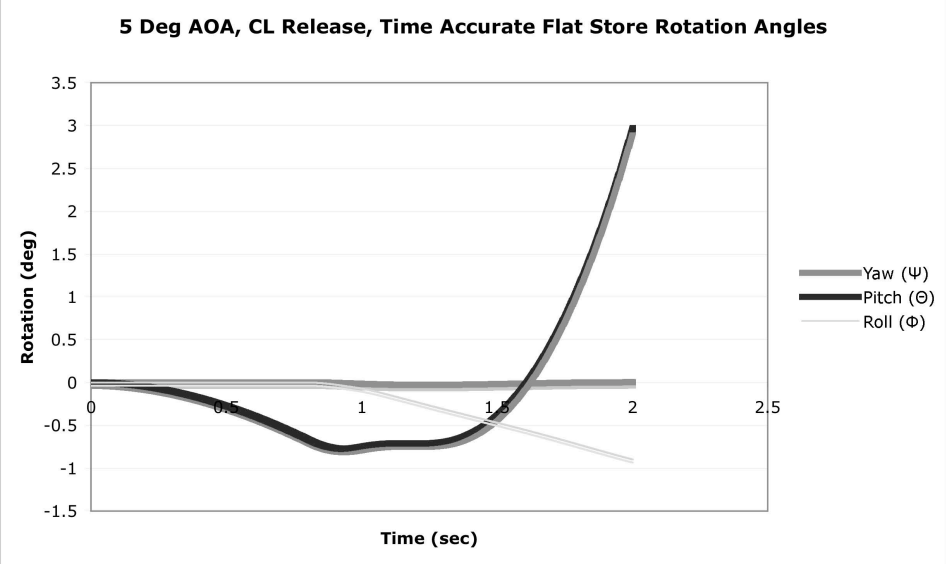


(b) Angular Displacement

Figure 25: Time Accurate Position and Angular Displacement of Flat Store as a Function of Time

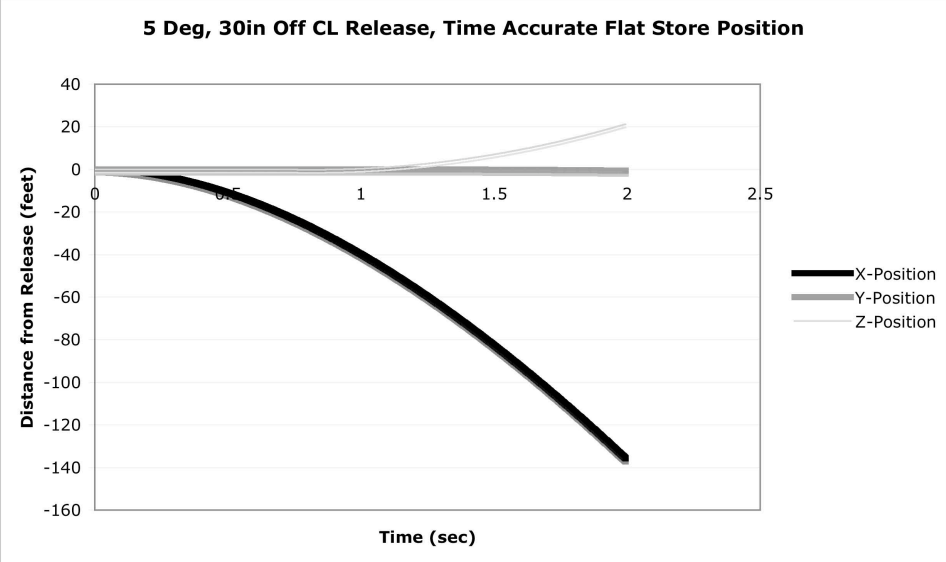


(a) Position

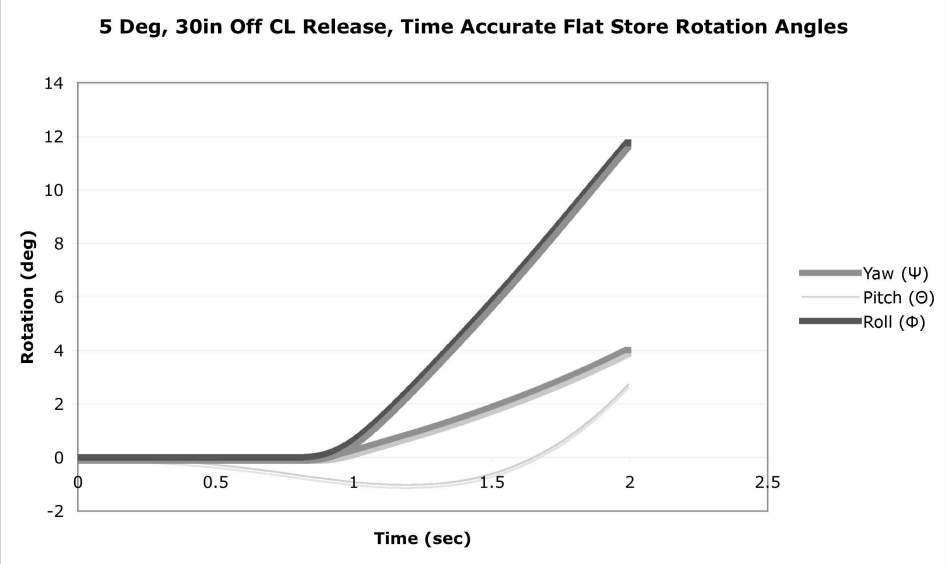


(b) Angular Displacement

Figure 26: Time Accurate, 5 Deg AOA CL Position and Angular Displacement of Flat Store Functions of Time



(a) Position



(b) Angular Displacement

Figure 27: Time Accurate, 5 Deg AOA Offset Position and Angular Displacement of Flat Store Functions of Time

difference from the earlier exit visualizations, but several shots have been included for completeness. In Figure 28 note that the store still continues to travel its safe arc down and away from the aircraft during the extraction, even at the greatest AOA and with the largest offset.

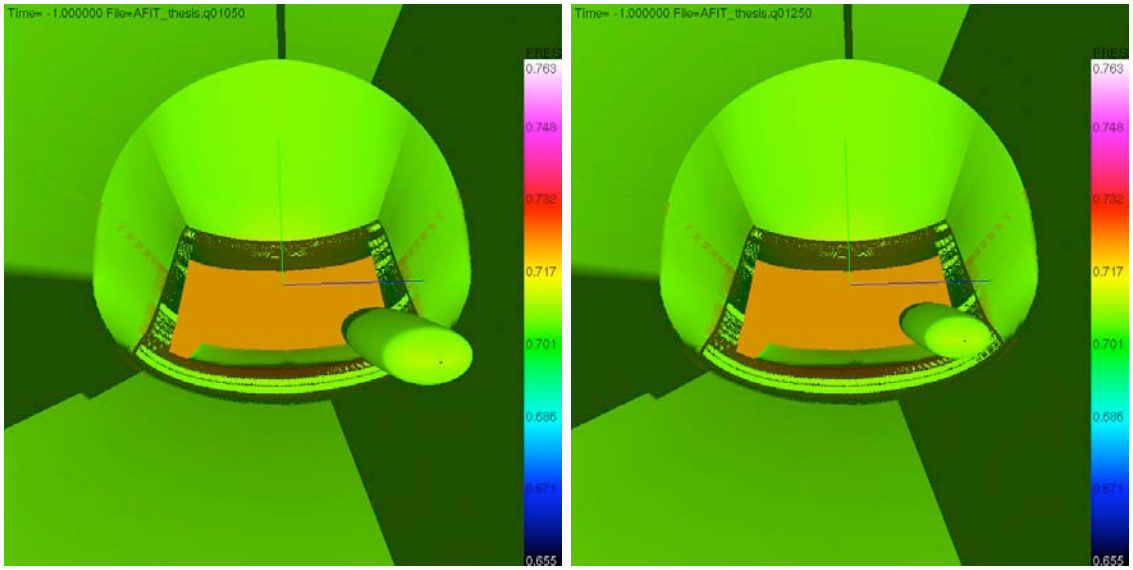
4.4 Results Overview

An examination of the last two runs in the series provides the best insight into an actual extraction of the desired store via a parachute. These two runs most accurately represent the conditions of a C-130 configured for airdrops with the appropriate AOA and a reasonably sound parachute force model. The updraft of air at the trailing edge of the cargo door is definitely present, but considering the magnitude of the force when coupled with the size and configurations of the store, it is not surprising that the store follows a safe arc away during the deployment. While the aerodynamic forces have been minimized due to the smooth edges and lack of control surfaces on the store models used in this analysis, there is no indication that a store of this size, when extracted by a parachute, will cause any follow-on danger to the C-130.

While there was some evidence, as present in the flow field visualizations, of the vortices generated by the aft end of the cargo door, the density of the grids at this point did not allow for a thorough investigation of their length, size and strength. This analysis was centered around the physical path that the store would follow once it left the cargo bay and entered the turbulent region aft of the aircraft. The far field grid generated to capture the path of the store and the surrounding flow field served its primary purpose of capturing the store's path, but would need to be significantly "beefed up" in the region of the aft end of the cargo door to capture the vortices with a greater fidelity.

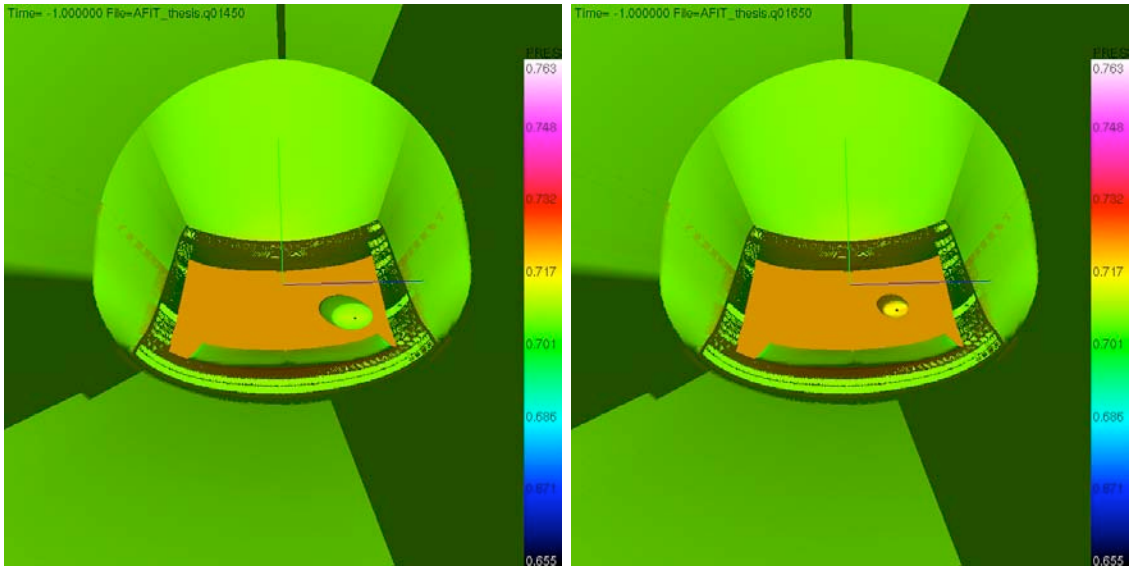
The author feels that integration of a full physics parachute model, use of the Lockheed proprietary JASSM grids and development of a launch rack that may be integrated into the existing analysis represent the most fruitful avenues of continued study. These suggestions will be explored in more depth in the Conclusions section.

As always, CFD is a simulation of actual events; prone to input errors, simplification and time limits for analysis. It represents an approximation of future flight test results and should be viewed with caution borne from solid engineering judgment. The "Way Ahead" and "Final thoughts" sections provide a general blueprint for follow-on work in this area.



(a) Start

(b) Plus 0.3 sec



(c) Plus 0.6 sec

(d) Plus 0.9 sec

Figure 28: Time Accurate Parachute Force Offset Flattened Store Exit Visualization

V. CONCLUSIONS

5.1 Summary

At the outset of this thesis, a quote from Sun Tzu was presented to frame the overall thrust of this research as it relates to the war fighter in the 21st century. It was understood that the goal was to march further down the path to the ability to strike a massive target set with surgical precision while maintaining overall threat exposure at its lowest practical level. This thesis represents one small step on one avenue of that path. It should in no way be understood that the author feels that this approach is the best way to solve the overarching problem, merely an explanation of the feasibility of this concept.

Having said that, it appears that the parachute extraction of a suitable CSW from the cargo bay of a C-130 is a viable proposition and merits further study as a proof of concept. The maximum pitch and roll displacements seen in the study never exceeded 13 degrees and more importantly, the store never showed a tendency to rise toward the empennage during the extraction while transiting the powerful vortices generated by the open cargo door.

Care should be given to understand the limitations of this analysis as applied to the real world as a result of the simplifying assumptions made – most notably, the parachute extraction model and dramatically simplified store geometry from the desired CSW. While the author believes that this is a valid "first-order solution" more extensive CFD analysis is required prior to any flight testing.

5.2 The Way Ahead

Given the state of the solution, the author has three chief recommendations for follow-on research that should be viewed as critical prior to any flight testing. They are: 1) Integration of a full physics model of a 15-foot parachute in the extraction configuration, 2) Make use of the proprietary JASSM grids for a greater fidelity of solution and finally, once the first two have been accomplished, 3) Development and integration of a launch rack model suitable for the multiple weapon release.

5.2.1 Parachute Model. The AFSEO CAT is currently working on a parachute model suitable for this problem. The grids for their parachute should be obtained and integrated into the solution. The ASC/ENFC division at WPAFB is an excellent source of information for actual airdrop parachute configuration. This information, when combined with the AFSEO parachute model should provide a time accurate solution for the forces on the store during the extraction.

5.2.2 JASSM Grids. Lockheed owns the grids for the JASSM model. Every avenue should be explored to obtain the legal right to utilize these grids and proceed with a dramatically higher fidelity model for this proof of concept. While the current generic store has the same mass properties, the author still feels the aerodynamic loading effects have been significantly under-represented in this analysis.

5.2.3 Launch Rack Development. The CSW will not be carried and launched as a single munition, as this defeats the purpose of using an aircraft capable of transporting a large number of weapons to the launch point. The concept of a "bank of filing cabinets" to contain the CSWs which is then extracted as a single unit, allowing for the sequential deployment of many weapons is not new. There are many current concepts in development along these lines. For this concept to become a reality, a suitable series of grids must be integrated into the existing model allowing for deployment of the mass of CSWs, as well as an analysis of the effects of this launching method on the range and endgame energy of the weapon as it reaches its target.

5.3 Final Thoughts

The concept of deploying a large number of these CSWs from a portion of our fleet of cargo aircraft is useful when examined without consideration for the associated hardware, aircrew training and force integration issues that arise with this concept of operations. However, prior to adoption of this mission, many larger issues must be addressed. A few are listed below.

5.3.1 Aircrew Training. The interdiction mission has little or no crossover with the current mission associated with this fleet of aircraft. Their aircrew are not currently supported in the manner necessary for adequate planning of a strike mission– from intelligence to mission planning. This mission is not something that should be attempted to be "folded into" existing tasking as another mission for aircrew to complete as a part of their Readiness Aircrew Program (RAP) squares. The Time Critical Targeting (TCT) mission requires extreme precision, minimal time (as the name implies) and absolute adherence to operating procedures. It cannot be picked up on the fly and only bad things can happen if this is attempted.

5.3.2 Programming Hardware. Integration of an extremely robust flight planning computer to develop flight profiles that may be "squirted" into the large number CSWs prior to their extraction is critical. These weapons are not a simple "fire and forget" proposition. Rather, they require extensive flight planning to hit their targets from the required axis, with the right energy and entry parameters. Their effectiveness is dramatically reduced when generic flight profiles are given to the munition. If we are going to the trouble to develop this capability, high fidelity on board planning should be integrated into the plan from its inception.

5.3.3 Radar Threat Detection on Cargo Aircraft. One of the primary suppositions in this research is that the delivery aircraft is largely outside the threat rings associated with the target area. Based on emerging threats, this assumption may not always be valid. Bearing this in mind, the author feels that prior to these aircraft assuming this mission, at minimum level of radar threat awareness should be instituted, assuming radar guided systems are the primary threat at the ranges in question. In addition, it is a gross simplification to assume aircraft outfitted with these weapons would seamlessly integrate with a larger war fighting effort. Without a more robust set of Radar Warning Receiver (RWR) gear on-board these aircraft, the mission commander will have to devote a disproportionate number of assets to protect or alert these delivery aircraft to incoming threats. Obviously, these mission

assets are not limitless and their allocation in a support role at the extremities of the battle would have a dramatic effect on front-line force protection.

Appendix A. Beggar Input File

What follows is a line-by-line review of the settings for the Beggar input file. This is the master input program which orders Beggar to assemble all grids into their respective super blocks, ensures grid to grid communication and establishes all the basic flow conditions and solver parameters. The commands are first explained and then the actual input file is attached for reference.

1. mach = 0.22, chosen for the nominal operational parameters from a typical air drop.(8). This rationale holds for the reynolds number, gravity, density and speed of sound.

2. rot z -5.0, this switch was enabled when the AOA was changed to 5 degrees in the later runs. Note that this actually adjusts the velocity vector of the free stream, to ensure the store still sees gravity in the appropriate orientation, the gravity must be broken up into an x and y component under the sixdof gravity "switch" or setting.

3. dt = 10, once again a recommendation from the AFSEO. In this case, dt, is the non-dimensional time step used in a time-accurate problem. This quantity is related to physical time (dt phys) through the following formula(1):

$$dt = \text{dt phys}\left(\frac{\text{sixdof soundspd}}{\text{sixdof reflength}}\right) \tag{6}$$

   The reference length is just a conversion between feet (the standard unit) and inches (standard aircraft station numbers that specifically defined the location of the cargo door in this problem). For this problem the physical time step was approximately $1.49x10^{-3}$ sec per iteration.

4. verbose = 3, this is a sliding scale from 0 (least amount of output) to 10 (most output), with 3 being the recommendation from the AFSEO as their standard diagnostic level for output(1)

5. ptol = 1e-5, which is the tolerance criteria at which point Beggar decides two points are the same. If their distance in three-space is less than or equal to the

value set, then the points are declared as a block-to-block interface. The Beggar manual cautions the user against using this parameter to solve errors in solution grids by setting this value too high. This is the default value in Beggar(1).

6. nopatch, sets grids to only block-to-block or overset communications during the assembly. One of the quirks of Beggar is that patched communications, if established during the initial assembly of grid and prior to flow solution, cause problems with the dynamic solver. As a result, this parameter is set, so that when the initial communications are set, patched grids are not allowed(7).

7. dtiter = 3, this parameter specifies the number of Newtonian, or dt iterations that are performed each time step for time accurate problems. This parameter is, again, the standard used by the AFSEO and provides a reasonable compromise between solution time and accuracy(1).

8. inner = 60, is the setting which specifies the number of Gauss-Seidel, or inner iterations used to solve the linearized system of governing equations(1). Basically, this is a stopping function in the case of a run-away solution. The default convergence criteria is four orders of magnitude and once that is achieved, the solver moves on to the next iteration. In the case of a solution that isn't converging, the inner setting stops the solver at the set parameter(7).

9. iter = 0, this setting tells the solver to only assemble the grids and establish communications. No flow solution is accomplished (1).

10. cfl = 30000, the Courant number is a standard stability criteria, used in the case of a time-accurate run to determine local time stepping which will increase the rate of convergence for a single time step(1). This is another experience-based parameter set with the concurrence of AFSEO.

11. limiter = vanalbada, sets the limiter function (ψ) for oscillations to the function proposed by van Albada, as shown in the following equation(17):

$$\psi(r) = \frac{r + |r|}{1 + r^2} \tag{7}$$

in which r is defined as:

$$r_{j+\frac{1}{2}} = \frac{u_{j+2} - u_{j+1}}{u_{j+1} - u_j} \tag{8}$$

and u is the dependent variable of interest at the same time step, from the current value through two steps upwind.

12. solver = , the options contained in this line setup the solver parameters and include calls for the following(1)(17):

    (a) second order–second order spatial accuracy

    (b) full–a full set of Jacobians on the left side of the equation

    (c) viscous baldwin-lomax–use of an algebraic Baldwin-Lomax algorithm for turbulence modeling

    (d) steg warm xair jacobians–Beggar has incorporated the XAIR coding for the Steger Warming Jacobians.

    (e) implicit bcupdate–while the inner, or Gauss-Seidel, iteration boundary conditions are solved explicitly, as these iterations coverage to the set tolerance they approach an implicit updating of the boundary conditions.

    (f) primitive extrap–while Beggar is a cell centered code, it uses a MUSCL scheme to extrapolate the primitive variables to the cell faces which are then used for flux calculations.

    (g) steger warming right side–Steger Warming is used for flow equation discretization.

    (h) implicit viscous i,j,k–calculation of the viscous terms with no cross derivatives, as well as the full viscous terms in the i,j and k are completed implicitly.

What follows is the actual text of the input file used in the Beggar code:

```
#...

    AC-130 input file for Beggar
```

```
...#
mach = 0.22
#rot z -5.0 # 5 deg AOA runs
dt=10
verbose = 3
ptol = 1e-5
nopatch
reynolds = 1140.0  # Rey/in, 5,000ft at Mach .22
dtiter=3
inner=60
iter = 0
cfl=30000
limiter=vanalbada
 solver=second order, full, viscous baldwin_lomax, steg_warm_xair jacobians,
        implicit bcupdate, primitive extrap,steger_warming right_side,
        implicit viscous_k, implicit viscous_j, implicit viscous_i

sixdof gravity = <0,-32.2,0>  #ft/s^2
#sixdof gravity = <2.8,-32.1,0>  #ft/s^2 gravity vector for 5 deg runs
sixdof density = 0.002378     #Slug/ft^3
sixdof soundspd = 1116.4      #ft/s
sixdof refl = 0.083333        #Reference length conversion
#init from 'restart'
######################################################################
#
# Read in C130 Grids
#
######################################################################
readgrids "/home/afit3/gae04s/splatt/Thesis/C130Package/grids/Farfield.p3ds"
as plot3d binary
include "/home/afit3/gae04s/splatt/Thesis/C130Package/beg/c130_fuse.beg"
readgrids "/home/afit3/gae04s/splatt/Thesis/C130Package/grids/
wing_s4_mirrd1.p3ds" as plot3d binary
tag "wing"  # sixteen blocks
  g 1
    set (1,1,1) (*,*,1) to tangent nocut     # fuselage
    set (1,1,1) (*,1,*) to tangent           # wing
  g 2
    set     (1,1,1) (*,*,1) to  tangent nocut      #  fuselage
    set     (1,1,1) (1,*,*) to  tangent            #  wing
  g 3
    set (1,1,1) (*,*,1) to tangent nocut     # fuselage
```

```
  set (1,1,1) (*,1,*) to tangent           # wing
  #set (1,*,1) (*,*,*) to overlap
g 4
  set     (1,1,1) (*,*,1) to  tangent nocut      #  fuselage
  set     (1,1,1) (1,*,*) to  tangent            #  wing
  #set (1,*,1) (*,*,*) to overlap
g 5
  set (1,1,1) (*,*,1) to tangent nocut     # fuselage
  set (1,1,1) (*,1,*) to tangent           # wing
g 6
  set     (1,1,1) (*,*,1) to  tangent nocut      #  fuselage
  set     (1,1,1) (1,*,*) to  tangent            #  wing
g 7
  set (1,1,1) (*,1,*) to tangent           # wing
g 8
  set     (1,1,1) (1,*,*) to  tangent            #  wing
g 9
  set (1,1,1) (*,*,1) to tangent nocut     # fuselage
  set (1,1,1) (*,1,*) to tangent           # wing
g 10
  set     (1,1,1) (*,*,1) to  tangent nocut      #  fuselage
  set     (1,1,1) (1,*,*) to  tangent            #  wing
g 11
  set (1,1,1) (*,1,*) to tangent           # wing
g 12
  set     (1,1,1) (1,*,*) to  tangent            #  wing
g 13
  set (1,1,1) (*,1,*) to tangent           # wing
g 14
  set     (1,1,1) (1,*,*) to  tangent            #  wing
g 15
  set (1,1,1) (*,1,*) to tangent           # wing
g 16
  set     (1,1,1) (1,*,*) to  tangent            #  wing
g 17
  set (1,1,1) (*,1,*) to tangent           # wing
g 18
  set     (1,1,1) (1,*,*) to  tangent            #  wing
g 19
  set (1,1,1) (*,1,*) to tangent           # wing
g 20
  set     (1,1,1) (1,*,*) to  tangent            #  wing
g 31
```

```
    set (1,1,1) (*,*,1) to tangent           # wing
  g 32
    set     (1,1,1) (*,*,1) to  tangent            #  wing
  g 33
    ptol = 0.001
    set (1,1,1) (*,*,1) to tangent           # wing
  g 34
    ptol = 0.001
    set     (1,1,1) (*,*,1) to  tangent            #  wing
readgrids "/home/afit3/gae04s/splatt/Thesis/C130Package/grids/
dflap_i_2an2_mirrd.p3ds" as plot3d binary
tag "flap_inb"
  g 5
    set (1,1,1) (*,1,*) to tangent          # lower surf
  g 6
    set     (1,1,1) (1,*,*) to  tangent           #  lower surf
  g 7
    set (1,1,1) (*,1,*) to tangent          # upper surf
  g 8
    set     (1,1,1) (1,*,*) to  tangent           #  upper surf
readgrids "/home/afit3/gae04s/splatt/Thesis/C130Package/grids/
dflap_o_2an2_mirrd.p3ds" as plot3d binary
tag "flap_outb"
  g 5
    set (1,1,1) (*,1,*) to tangent          # lower surf
  g 6
    set     (1,1,1) (1,*,*) to  tangent           #  lower surf
  g 7
    set (1,1,1) (*,1,*) to tangent          # upper surf
  g 8
    set     (1,1,1) (1,*,*) to  tangent           #  upper surf
readgrids "/home/afit3/gae04s/splatt/Thesis/C130Package/grids/
wing_flap_int12a_mirrd.p3ds" as plot3d binary
tag "flap_int12"
  g 1
    set (1,*,1) (*,*,*) to tangent nocut
    set (1,1,1) (*,*,*) to protect
  g 2
    set     (*,1,1) (*,*,*) to  tangent nocut
    set     (1,1,1) (*,*,*) to  protect
  g 3
    set (1,1,1) (*,*,*) to protect
  g 4
```

```
  set     (1,1,1) (*,*,*) to  protect
g 5
  set (1,*,1) (*,*,*) to tangent nocut
  set (1,1,1) (*,*,*) to protect
g 6
  set     (*,1,1) (*,*,*) to  tangent nocut
  set     (1,1,1) (*,*,*) to  protect
g 7
  set (1,1,1) (*,1,*) to tangent nocut
  set (1,1,1) (*,*,*) to protect
g 8
  set     (1,1,1) (1,*,*) to  tangent nocut
  set     (1,1,1) (*,*,*) to  protect
g 9
  set (1,1,1) (*,*,*) to protect
g 10
  set     (1,1,1) (*,*,*) to  protect
g 11
  set (1,1,1) (*,1,*) to tangent nocut
  set (1,1,1) (*,*,*) to protect
g 12
  set     (1,1,1) (1,*,*) to  tangent nocut
  set     (1,1,1) (*,*,*) to  protect
g 13
  set (1,*,1) (*,*,*) to tangent nocut
  set (1,1,1) (*,*,*) to protect
g 14
  set     (*,1,1) (*,*,*) to  tangent nocut
  set     (1,1,1) (*,*,*) to  protect
g 15
  set (1,1,1) (*,*,*) to protect
g 16
  set     (1,1,1) (*,*,*) to  protect
g 17
  set (1,*,1) (*,*,*) to tangent nocut
  set (1,1,1) (*,*,*) to protect
g 18
  set     (*,1,1) (*,*,*) to  tangent nocut
  set     (1,1,1) (*,*,*) to  protect
g 19
  set (1,1,1) (*,1,*) to tangent nocut
  set (1,1,1) (*,*,*) to protect
g 20
```

```
    set     (1,1,1) (1,*,*) to  tangent nocut
    set     (1,1,1) (*,*,*) to  protect
  g 21
    set (1,1,1) (*,*,*) to protect
  g 22
    set     (1,1,1) (*,*,*) to  protect
  g 23
    set (1,1,1) (*,1,*) to tangent nocut
    set (1,1,1) (*,*,*) to protect
  g 24
    set     (1,1,1) (1,*,*) to  tangent nocut
    set     (1,1,1) (*,*,*) to  protect
readgrids "/home/afit3/gae04s/splatt/Thesis/C130Package/grids/
wing_flap_intext4_mirrd.p3ds" as plot3d binary
tag "flap_intext4"
  g 1
    set (1,*,1) (*,*,*) to tangent nocut
    set (1,1,1) (*,*,*) to protect
  g 2
    set     (*,1,1) (*,*,*) to  tangent nocut
    set     (1,1,1) (*,*,*) to  protect
  g 3
    set (1,1,1) (3,*,*) to protect
  g 4
    set     (1,1,1) (*,3,*) to  protect
  g 5
    set (1,1,1) (*,1,*) to tangent nocut
    set (1,1,1) (*,*,*) to protect
  g 6
    set     (1,1,1) (1,*,*) to  tangent nocut
    set     (1,1,1) (*,*,*) to  protect
  g 7
    set (1,1,1) (3,*,*) to protect
  g 8
    set     (1,1,1) (*,3,*) to  protect
readgrids "/home/afit3/gae04s/splatt/Thesis/C130Package/grids/
wing_flap_int4_mirrd.p3ds" as plot3d binary
tag "flap_int4"
  g 1
    set (1,*,1) (*,*,*) to tangent nocut
    set (1,1,1) (*,*,*) to protect
  g 2
    set     (*,1,1) (*,*,*) to  tangent nocut
```

```
    set     (1,1,1) (*,*,*) to  protect
  g 3
    set (1,1,1) (3,*,*) to protect
  g 4
    set     (1,1,1) (*,3,*) to  protect
  g 5
    set (1,1,1) (*,1,*) to tangent nocut
    set (1,1,1) (*,*,*) to protect
  g 6
    set     (1,1,1) (1,*,*) to  tangent nocut
    set     (1,1,1) (*,*,*) to  protect
  g 7
    set (1,1,1) (3,*,*) to protect
  g 8
    set     (1,1,1) (*,3,*) to  protect
readgrids "../grids/wing_fuse_dflapi2_gap_mirrd.p3ds"
as plot3d binary
tag "wing_fuse_flap_gap"
  g 1
    set (1,1,1) (1,*,*) to tangent nocut
    set (1,1,1) (*,*,1) to tangent nocut
    set (1,1,1) (*,*,*) to protect
  g 2
    set     (1,1,1) (*,1,*) to  tangent nocut
    set     (1,1,1) (*,*,1) to  tangent nocut
    set     (1,1,1) (*,*,*) to  protect
  g 3
    set (1,1,1) (*,*,1) to tangent nocut
    set (1,1,1) (*,*,*) to protect
  g 4
    set     (1,1,1) (*,*,1) to  tangent nocut
    set     (1,1,1) (*,*,*) to  protect
  g 5
    set (1,1,1) (1,*,*) to tangent nocut
    set (1,1,1) (*,*,1) to tangent nocut
    set (1,1,1) (*,*,*) to protect
  g 6
    set     (1,1,1) (*,1,*) to  tangent nocut
    set     (1,1,1) (*,*,1) to  tangent nocut
    set     (1,1,1) (*,*,*) to  protect
  g 7
    set (1,1,1) (*,*,1) to tangent nocut
    set (1,1,1) (*,*,*) to protect
```

```
g 8
  set     (1,1,1) (*,*,1) to  tangent nocut
  set     (1,1,1) (*,*,*) to  protect
g 9
  set (1,1,1) (*,*,1) to tangent nocut
  set (1,1,1) (*,1,*) to tangent nocut
  set (1,1,1) (*,*,*) to protect
g 10
  set     (1,1,1) (*,*,1) to  tangent nocut
  set     (1,1,1) (1,*,*) to  tangent nocut
  set     (1,1,1) (*,*,*) to  protect
g 11
  set (1,1,1) (*,*,1) to tangent nocut
  set (1,1,1) (*,*,*) to protect
g 12
  set     (1,1,1) (*,*,1) to  tangent nocut
  set     (1,1,1) (*,*,*) to  protect
g 13
  set (1,1,1) (*,*,1) to tangent nocut
  set (1,1,1) (*,*,*) to protect
g 14
  set     (1,1,1) (*,*,1) to  tangent nocut
  set     (1,1,1) (*,*,*) to  protect
g 15
  set (1,1,1) (1,*,*) to tangent nocut
  set (1,1,1) (*,*,*) to protect
g 16
  set     (1,1,1) (*,1,*) to  tangent nocut
  set     (1,1,1) (*,*,*) to  protect
g 17
  set (1,*,1) (*,*,*) to tangent nocut
  set (1,1,1) (*,*,*) to protect
g 18
  set     (*,1,1) (*,*,*) to  tangent nocut
  set     (1,1,1) (*,*,*) to  protect
g 19
  set (1,1,1) (*,1,*) to tangent nocut
  set (1,1,1) (*,*,*) to protect
g 20
  set     (1,1,1) (1,*,*) to  tangent nocut
  set     (1,1,1) (*,*,*) to  protect
g 21
  set (1,1,1) (*,*,*) to protect
```

```
  g 22
    set     (1,1,1) (*,*,*) to  protect
  g 23
    set (1,1,1) (*,1,*) to tangent nocut
    set (1,1,1) (*,*,*) to protect
  g 24
    set     (1,1,1) (1,*,*) to  tangent nocut
    set     (1,1,1) (*,*,*) to  protect
readgrids "/home/afit3/gae04s/splatt/Thesis/C130Package/grids/
eng_i2_mirrd.p3ds" as plot3d binary
tag "inboard engine"
  g 1
    set (1,1,1) (*,1,*) to tangent
    set (*,1,1) (*,*,*) to tangent nocut     # wing surface
    set (*-1,1,1) (*,*,*) to protect
  g 2
    set     (1,1,1) (1,*,*) to  tangent
    set     (1,*,1) (*,*,*) to  tangent nocut      #  wing surface
    set     (1,*-1,1) (*,*,*) to  protect
  g 3
    set (1,1,1) (*,1,*) to tangent
    set (*,1,1) (*,*,*) to tangent nocut     # wing surface
    set (1,1,1) (*,*,*) to protect
  g 4
    set     (1,1,1) (1,*,*) to  tangent
    set     (1,*,1) (*,*,*) to  tangent nocut      #  wing surface
    set     (1,1,1) (*,*,*) to  protect
  g 5
    set (1,1,1) (*,1,*) to tangent
    set (1,1,1) (*,*,*) to protect
  g 6
    set     (1,1,1) (1,*,*) to  tangent
    set     (1,1,1) (*,*,*) to  protect
  g 7
    set (1,1,1) (*,1,*) to tangent nocut
    set (1,1,1) (*,*,1) to tangent nocut     # wing surface
    set (1,1,*) (*,*,*) to tangent nocut     # wing surface
    set (1,1,1) (*,*,*) to protect
  g 8
    set     (1,1,1) (1,*,*) to  tangent nocut
    set     (1,1,1) (*,*,1) to  tangent nocut      #  wing surface
    set     (1,1,*) (*,*,*) to  tangent nocut      #  wing surface
    set     (1,1,1) (*,*,*) to  protect
```

```
readgrids "/home/afit3/gae04s/splatt/Thesis/C130Package/grids/
eng_02_mirrd.p3ds" as plot3d binary
tag "outboard engine"
  g 1
    set (1,1,1) (*,1,*) to tangent
    set (*,1,1) (*,*,*) to tangent nocut     # wing surface
    set (1,1,1) (*,*,*) to protect
  g 2
    set     (1,1,1) (1,*,*) to  tangent
    set     (1,*,1) (*,*,*) to  tangent nocut      #  wing surface
    set     (1,1,1) (*,*,*) to  protect
  g 3
    set (1,1,1) (*,1,*) to tangent
    set (*,1,1) (*,*,*) to tangent nocut     # wing surface
    set (*-1,1,1) (*,*,*) to protect
  g 4
    set     (1,1,1) (1,*,*) to  tangent
    set     (1,*,1) (*,*,*) to  tangent nocut      #  wing surface
    set     (1,*-1,1) (*,*,*) to  protect
  g 5
    set (1,1,1) (*,1,*) to tangent
    set (1,1,1) (*,*,*) to protect
  g 6
    set     (1,1,1) (1,*,*) to  tangent
    set     (1,1,1) (*,*,*) to  protect
  g 7
    set (1,1,1) (*,1,*) to tangent nocut
    set (1,1,1) (*,*,1) to tangent nocut     # wing surface
    set (1,1,*) (*,*,*) to tangent nocut     # wing surface
    set (1,1,1) (*,*,*) to protect
  g 8
    set     (1,1,1) (1,*,*) to  tangent nocut
    set     (1,1,1) (*,*,1) to  tangent nocut      #  wing surface
    set     (1,1,*) (*,*,*) to  tangent nocut      #  wing surface
    set     (1,1,1) (*,*,*) to  protect
readgrids "/home/afit3/gae04s/splatt/Thesis/C130Package/grids/
tank_a2_mirrd.p3ds" as plot3d binary
tag "tank"
  g 1
    set (1,1,1) (*,1,*) to tangent
  g 2
    set     (1,1,1) (1,*,*) to  tangent
  g 3
```

```
    set (1,1,1) (*,1,*) to tangent
  g 4
    set     (1,1,1) (1,*,*) to  tangent
  g 5
    set (1,1,1) (*,1,*) to tangent
  g 6
    set     (1,1,1) (1,*,*) to  tangent
  g 7
    set (1,1,1) (*,1,*) to tangent
  g 8
    set     (1,1,1) (1,*,*) to  tangent
readgrids "/home/afit3/gae04s/splatt/Thesis/C130Package/grids/
tank_pylon_a3_mirrd.p3ds" as plot3d binary
tag "tank pylon"
  g 1
    set (1,1,1) (*,*,*) to protect
    set (1,1,1) (*,1,*) to tangent           # pylon
    set (1,1,1) (*,*,1) to tangent nocut     # wing
    set (1,1,*) (*,*,*) to tangent nocut     # tank
  g 2
    set     (1,1,1) (*,*,*) to  protect
    set     (1,1,1) (1,*,*) to  tangent            #  pylon
    set     (1,1,1) (*,*,1) to  tangent nocut      #  wing
    set     (1,1,*) (*,*,*) to  tangent nocut      #  tank
  g 3
    set (1,1,1) (*,*,*) to protect
    set (1,1,1) (*,1,*) to tangent           # pylon
    set (1,1,1) (*,*,1) to tangent nocut     # wing
    set (1,1,*) (*,*,*) to tangent nocut     # tank
  g 4
    set     (1,1,1) (*,*,*) to  protect
    set     (1,1,1) (1,*,*) to  tangent            #  pylon
    set     (1,1,1) (*,*,1) to  tangent nocut      #  wing
    set     (1,1,*) (*,*,*) to  tangent nocut      #  tank
readgrids "/home/afit3/gae04s/splatt/Thesis/C130Package/grids/
blast_area_mirrd.p3ds" as plot3d binary
tag "blast area"
  g 1
    set (1,1,1) (*,1,*) to tangent nocut     # fuselage
    set (1,1,1) (*,*,1) to tangent nocut     # wing
  g 2
    set     (1,1,1) (1,*,*) to  tangent nocut      #  fuselage
    set     (1,1,1) (*,*,1) to  tangent nocut      #  wing
```

```
  g 3
    set (1,1,1) (*,1,*) to tangent nocut     # fuselage
    set (1,1,1) (*,*,1) to tangent nocut     # wing
  g 4
    set     (1,1,1) (1,*,*) to  tangent nocut      #  fuselage
    set     (1,1,1) (*,*,1) to  tangent nocut      #  wing
  g 5
    set (1,1,1) (*,1,*) to tangent nocut     # fuselage
  g 6
    set     (1,1,1) (1,*,*) to  tangent nocut      #  fuselage
  g 7
    set (1,1,1) (*,*,1) to tangent nocut     # wing
  g 8
    set     (1,1,1) (*,*,1) to  tangent nocut      #  wing
  g 9
    set (1,1,1) (*,*,1) to tangent nocut     # wing
  g 10
    set     (1,1,1) (*,*,1) to  tangent nocut      #  wing

readgrids "../grids/dropgrid.p3ds" as plot3d binary
include "/home/afit3/gae04s/splatt/Thesis/C130Package/beg/doorramp.beg"
include "/home/afit3/gae04s/splatt/Thesis/C130Package/beg/cargo_int.beg"
include "/home/afit3/gae04s/splatt/Thesis/C130Package/beg/aft_int.beg"
include "/home/afit3/gae04s/splatt/Thesis/C130Package/beg/store.beg"
include "/home/afit3/gae04s/splatt/Thesis/C130Package/R0/store.fspec"
include "/home/afit3/gae04s/splatt/Thesis/C130Package/R0/store.dyn"
include "/home/afit3/gae04s/splatt/Thesis/C130Package/R0/door.fspec"
#_______________________________________________________________________________
#
#                SET WORLDSIDE FOR CCUT OPTIONS
#_______________________________________________________________________________
#
sb 1
 g 1
  set (1,1,1) (2,2,2) to worldside
```

Appendix B. Beggar Grid Input Files

B.1 Store inputs

B.1.1 Grid Input. What follows is the Beggar input file used to read the store grid and scale it for subsequent runs.

```
readgrids '../grids/store_body2.p3ds' as plot3d binary
tag 'storebody_SB'
#scaley 0.65
  g 1
    init (1,1,1) (*,*,*) using mach=0.01
    set "storebody"  =  (1,1,1) (*,*,1) to noslip
  g 2
    init (1,1,1) (*,*,*) using mach:q=0.01
    set "storebody" +=  (1,1,1) (*,*,1) to noslip
```

The grid is read in and tagged for reference in the event output is desired. The most important note is the scale command. The "scaley" command reduces the height of the store to 65 percent of it's original height. This was the method used to squash or "flatten" the store. The power of this command is that the user is able to use the original grids without modification and merely adjust their shape on a specified axis. Note also that the boundary condition was set to "no slip" for the viscous store. This is the only grid in the construction that was set for a viscous solution. This is also the most common method of solution for the AFSEO CAT when they are solving store separation problems.

B.1.2 Dynamic Specification. This next file is used to declare the inertial properties of the store as well as apply the parachute force for the extraction.

```
#==============================
#  Dynamic spec (store)
#==============================
dynamicspec "store_DG":
                        add fspec 'store_FS';
                        add sb 'storebody_SB';
#
 store main body with translational constraint;
  vector along smb translation direction = <1,0,0>;
  vector along smb axis of rotation = <0,0,1>;
  displacement equivalent to a degree of angular displacement = 40.;
  constraint release time = 0.8;
 mass = 67.87;
 ixx = 23.4; iyy = 653.9; izz = 650.0;
 ixy = -0.6; iyz = 0.00; ixz = 0.00;
 cg = <6.32, 0.042, 0.0042>;

           trans  400.0,-15.0,0.0;   # forward in bay on centerline
#          trans  400.0,-15.0,30.0;   # left side test
#          trans  700.0,-15.0,30.0;   # half out door left side

                        trelease = 0.0;
#
                        dump idaps left release;
                        dump gandc z down;
#Parachute Force
 add force with
  duration = 100.0,
#  components = <6150.0,0.0,0.0>,
  components = <1.0,0.0,0.0>,
  magnitude = func[0.1153*((1/(0.001485*t+0.004232))**2)],
  location   = <6.32, 0.042, 0.0042>,
  system = local
```

This file designates an initial constraint that holds the store the proper height above the floor of the cargo bay until it exits the aircraft. This constraint is meant to simulate having the store on a delivery sled of some type that would allow for the programming of the store as well as holding it for loading into the aircraft. The assumption is made here that once the store exits the bay, the store would be released

from the sled to continue the extraction. The 0.8 seconds that this constraint was in effect was a trial and error value arrived at after a number of test runs(7).

The mass and inertial properties were pulled directly from the Lockheed data for the JASSM, as well as the CG location. The translation commands allow the store to be placed in various locations in the cargo bay. The trelease command sets the time until the store is released–immediately in this case.

The parachute force is applied in the last section. The first commented out section has an x-component of 6150, which is the peak chute force used for the first runs. After the clearance was verified, the more accurate model was implemented. In this case, the x-component is reduced and the magnitude for the multiplier is derived from the the function. The inputted function is a simplification of the time accurate parachute force equation with all the values entered. The t in the function is the actual physical time at each iteration and is a standard Beggar reference derived for just such an application. The local system refers to the local system of the store in which the parachute force is applied. As discussed in the main body of the text, this simplification does not capture oscillations in the axis of the force as the parachute moves in the wind stream, but is a reasonable approximation until the full parachute model is incorporated.

B.2 Cargo Door

B.2.1 Cargo Door Grid Input. This file reads the grids in and sets the boundary condition such that that flow stream must pass around the door.

```
readgrids '../rampdoorgrids/doordown1a.p3ds' as plot3d binary
  tag 'C130_door_SB'
ptol=.001
  g 1
    set "RampDoor"  =  (22,26,*) (48,71,*) to tangent
  g 2
    set "RampDoor" +=  (22,26,1) (48,71,1) to tangent
```

The "ptol" command has already been discussed, and is used to declare points of within 0.001 inches in three space as the same point. This command was instituted to merge the points of close tolerance together declaring a block-to-block interface once the grids had been mirrored to the right side of the aircraft. The tangent command is used to establish the location of the door inside the grid volume.

B.2.2 Cargo Door Force Specification. As discussed in the methodology section, it was important to "initialize" the flow field prior to beginning the extraction. To that end, the forces were non-dimensionalized on the door using the syntax contained in the Beggar manual (1). Note that the length of the door and the approximate cross-sectional area were used. In addition, a point was chosen to evaluate the forces, which is specified in the line "mcenter" in the global coordinate system. These parameters are not exact, merely a tool obtain the proper order of magnitude. Figure 16 in the chapter 4 illustrates the point. Below is the input file:

```
# Force spec
forcespec "door_FS": dump every 1 to "door_forces.dat" with noheader
 with refl=  132.0  #Approx lenght of door
 with refa= 15840.0  #Approx cross sectional area using length and width
 with mcenter = <803.0,0.0,10.0> #
forcespec "door_FS": add "RampDoor"
```

B.3 Fuselage

B.3.1 Exterior Fuselage. The exterior of the C-130 was generated from the modified and mirrored baseline grids. As mentioned in the body of the text, Mr. Robert Moran and Mr. Bruce Jolly, of the AFSEO CAT were invaluable in this effort. Below is the text of the input file:

```
readgrids "../grids/fuse_a_mirrd1a.p3ds" as plot3d binary
 tag 'c130body_SB'
  g 1
    set "fuse"  =  (1,1,1) (*,1,*) to tangent
  g 2
    set 'fuse_m'  = (1,1,1) (1,*,*) to  tangent
  g 3
    set  (46,1,1) (60,1,55) to tangent
    set  (60,1,1) (85,1,*) to tangent
    set  (1,1,1) (46,1,*) to tangent
    set "rudder"  =  (65,1,1) (85,*,1) to tangent
  g 4
    set     (1,46,1) (1,60,55) to  tangent
    set     (1,60,1) (1,85,*) to  tangent
    set     (1,1,1) (1,46,*) to  tangent
    set 'rudder_m'  = (1,65,1) (*,85,1) to  tangent
  g 13
    init (1,1,1) (*,*,*) using mach=0.01
    set  (1,1,*) (*,*,*) to tangent
    set  (*,1,1) (*,*,*) to tangent
    set  (1,1,1) (1,*,*) to tangent
    set  (1,1,1) (*,1,*) to tangent
    set  (1,*,1) (*,*,*) to  tangent
  g 14
    init (1,1,1) (*,*,*) using mach=0.01
    set     (1,1,1) (5,*,1) to  tangent nocut
    set     (41,1,1) (*,*,1) to  tangent nocut
    set     (5,1,1) (41,5,1) to  tangent nocut
    set     (5,20,1) (41,*,1) to  tangent nocut
    set     (1,1,1) (*,1,*) to  tangent
    set     (1,*,1) (*,*,*) to  tangent
    set     (*,1,1) (*,*,*) to  tangent
    set     (1,1,1) (1,*,*) to  tangent
    set  (1,1,1) (*,7,3) to protect
```

The syntax for reading the grids in and designating the super block that contains the fuselage is by now familiar to the reader. The variety of "tangent" and "tangent nocut" conditions establish the boundary conditions necessary to properly generate the fuselage in computational space. The "nocut" condition stops this surface from cutting into other surfaces during the assembly and is usually utilized when there are two or more overlapping surfaces(1). The "init" command is used by Beggar to

initialize flow variables to a specified pressure, density and/or Mach number(1). In this case, the "init" command is used to ease the transition in the flow field around the cargo void and the flow is initialized from "slug flow" to the stabilized flow when the extraction is initiated. Often, if the flow isn't initialized to a value that is close to reality, there will be significant problems with stability during the start up of the solution(7).

B.3.2 Cargo Bay. The cargo void was inserted with the following file:

```
readgrids "../grids/cargo_int.p3ds" as plot3d binary
  tag 'cargo_int_SB'
  g 1
    init (1,1,1) (*,*,*) using mach=.01
    set     (1,1,1) (*,1,*) to  tangent nocut
    set     (1,1,*) (*,*,*) to  tangent nocut
    set     (*,1,1) (*,*,*) to  tangent nocut
    set     (1,1,1) (1,*,*) to  tangent nocut
```

The initialization and tangent nocut commands have been discussed and serve to establish the void at the proper conditions so that the store has a representative flow field transition to experience during the exit.

B.4 Interface Grids

As discussed in the main body, the use of interface grids is not the most elegant solution to the problem, but one that is often relatively quick to implement. As a result, when interface between super blocks became problematic, these grids were added. The input file follows:

```
readgrids "/home/afit3/gae04s/splatt/Thesis/C130Package/grids/aft_int3.p3ds"
as plot3d binary
  tag 'aft_int_SB'
  g 1
    set  (1,1,1) (20,*,1) to  tangent nocut
    set  (54,1,1) (*,*,1) to  tangent nocut
    set  (20,1,1) (54,10,1) to  tangent nocut
    set  (20,46,1) (54,*,1) to  tangent nocut
    set  (1,1,1) (18,*,7) to protect
    set  (1,44,1) (*,*,7) to protect
    set  (1,1,1) (*,14,7) to protect
    set  (53,1,1) (*,*,6) to protect
  g 2
    set  (1,1,*-5) (5,*,*) to protect
    set  (34,1,*-5) (*,*,*) to protect
    set  (1,1,*-5) (*,5,*) to protect
    set  (1,*-5,*-5) (*,*,*) to protect
readgrids "/home/afit3/gae04s/splatt/Thesis/C130Package/grids/huge3a.p3ds"
as plot3d binary
  g 1
    set  (1,1,1) (9,*,1) to  tangent nocut
    set  (9,1,1) (*,5,1) to  tangent nocut
    set  (9,48,1) (*,*,1) to tangent nocut
    set  (1,1,1) (*,*,5) to protect
  g 2
    set  (1,1,*-4) (*,*,*) to protect
readgrids "/home/afit3/gae04s/splatt/Thesis/C130Package/rampdoorgrids/
backend.p3ds" as plot3d binary
```

The first two grids utilized covered the interface between the cargo door front end and the fuselage (grids "aft int" and "huge3a"). The last grid ("backend") covered the trailing edge of the cargo door as the bomb nose passed by during the latter stages of the extraction.

Bibliography

1. AFSEO Computational Aeromechanics Team. *Beggar Version 106 User's Manualr*, 2004.
2. Air Force Seek Eagle Office Computational Aeromechanics Team. *Analysis of JASSM Release from a B-52*. Technical Report, Eglin AFB, FL: Air Force Seek Eagle Office, 2004.
3. Blazek, J. *Computational Fluid Dynamics: Principles and Applications*. Elsevier, 2001.
4. Drab, Capt. Jess and Bruce C. Patterson. "Massive Ordnance Air Blast Weapon Development," *AFRL Tech Horizons*, 14–16 (2004).
5. Franke, M. E. and Gurler Air. "Use of Cargo Aircraft for Launching Precision-Guided Munitions." Number 2004-1250 in 42nd AIAA Aerospace Sciences Meeting and Exhibit. Reno, NV: AIAA, 5-8 January 2004.
6. Johnson, William III, et al. "Experimental and Computational Investigation of the Flow Behind a C-130 with Tailgate Down." Number 2002-0713. AIAA, 2002.
7. Jolly, Bruce A. TEAS Task Leader, Jacobs Sverdrup, Computational Aeromechanics Team, Air Force SEEK EAGLE Office, Eglin AFB. Telephone Interview, 10 February 2005.
8. Kuntavanish, Mark A. "Airdrop Data Binder." personal notes.
9. Kuntavanish, Mark A. "Low Velocity Air Drop, Appendix A." personal notes.
10. Lijewski, Lawrence E. and Dave M. Belk. *Generation of a Multi-Component Grid System Using NGP and Beggar*. Technical Report, Wright Laboratory Armament Directorate Eglin AFB, FL, 2000.
11. Maple, Raymond C. "Aero 543 Advanced Computational Modeling for Aerodynamics." Class Notes, 1 Nov, 2004.
12. Martin, Kenneth L., et al. *Evaluation of the C-130 Stability and Control Characteristics and the A/A32H-4 Dual Rail Cargo Handling System During Low Level Cargo Deliveries* (First Edition). Technical Report, 1967.
13. Riess, George W. *Feasibility Study of ICBM Parachute Extraction From a C-5 Galaxy Cargo Bay*. Technical Report, Wright Patterson AFB, OH: Air Force C-5 Systems Program Office, 24 October 1974.
14. Rizk, Magdi, et al. "Beggar - A Store Separation Predictive Tool." Number 2002-3190 in 32nd AIAA Fluid Dynamics Conference and Exhibit. St. Louis, MO: AIAA, 24-26 June 2003.

15. Serrano, Matthieu, et al. "Computational Aerodynamics of the C-130 in Airdrop Configurations." Number 2003-0229 in Aerospace Sciences Meeting. Reno, NV: AIAA, 2003.

16. Space and Missile Systems Organization (SAMSO). "Modified Minuteman Launched from C-5." Publicity Package, 24 Oct 1974.

17. Tannehill, John C., et al. *Computational Fluid Mechanics and Heat Transfer*. Philadelphia, PA: Taylor and Francis, 1997.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
21–03–2005	Master's Thesis	Sep 2003 — Mar 2005

4. TITLE AND SUBTITLE

Parachute Extraction of a Generic Store from a C-130; a CFD Proof of Concept

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Platt, Stephen C., MAJ, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GAE/ENY/05-M17

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This thesis encompasses a feasibility analysis of a parachute extracted generic precision guided munition from the cargo bay of a C-130 aircraft in flight. This analysis utilizes the USAF Beggar code and incorporates full physics effects as well as aerodynamic loading assuming an inviscid aircraft and viscous store for a time-accurate solution. Both an immediate and time varying application of the parachute force are utilized as well as two different ordnance body styles at zero and 5 degrees AOA with the store placed on centerline and offset in the cargo bay. The time accurate parachute model is based on empirical data and more closely follows the force fall off as the parachute slows down during the extraction process. Both store body styles were successfully extracted from the cargo bay without contacting any portion of the delivery aircraft, following a safe trajectory down and away from all of the release conditions. The extraction took 1.7 seconds with the immediate application of the parachute force and 2.1 seconds when the time varying model was applied. The maximum roll seen during an extraction was 13 degrees, which was the largest movement on any axis.

15. SUBJECT TERMS

C-130, Parachute Extraction, JASSM, Beggar

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Milton E. Franke, USAF (ENY)
U	U	U	UU	87	19b. TELEPHONE NUMBER *(include area code)* (937) 255–3636, ext 4720